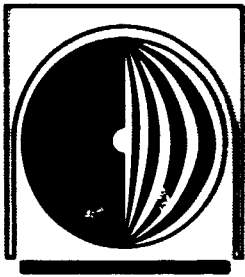


## **General Disclaimer**

### **One or more of the Following Statements may affect this Document**

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.



**DIVISION OF  
SOLID MECHANICS  
STRUCTURES  
AND  
MECHANICAL DESIGN**

**REPORT NO. 24**

**AN ALGORITHM FOR THE  
MINIMUM WEIGHT DESIGN  
OF THE GENERAL TRUSS**

**BY**

**MILTON J. SCHRADER**

**JUNE 1968**

**SCHOOL OF ENGINEERING • CASE WESTERN RESERVE UNIVERSITY**

FAULTY FORM 602

N69-27094	(THRU)
(ACCESSION NUMBER)	(CODE)
136	32
(PAGES)	(CATEGORY)
CR-66289	
(NASA CR OR TMX OR AD NUMBER)	

Program supported in part by the National Aeronautics and Space  
Administration Under Research Grant No. NsG 110-61

AN ALGORITHM FOR THE MINIMUM WEIGHT  
DESIGN OF THE GENERAL TRUSS

Report No. 24

by

Milton J. Schrader

June 1968

Approved by:

Richard L. Fox  
Assistant Professor  
of Engineering

SOLID MECHANICS,  
STRUCTURES AND  
MECHANICAL DESIGN

## FOREWORD

This report is a reproduction of a thesis submitted to Case Western Reserve University in partial fulfillment of the requirements for the Degree of Masters of Science.

This work is part of the continuing effort in the application of structural synthesis to aerospace vehicle structures supported in part by the National Aeronautics and Space Administration under Grant No. NsG 110-61.



An Algorithm for the Minimum Weight Design  
of the General Truss

by

Milton James Schrader

ABSTRACT

An algorithm is presented for the minimum weight design of a three dimensional, linear truss. The truss has an arbitrary number of nodes, members, and imposed load conditions. The members are assumed to be tubular and are described by the mean diameter and wall thickness. The truss is subjected to constraints on the nodal displacements, member stresses, and member sizes. More specifically, the stresses are limited by the yield strength of the material and the crippling and Euler buckling stress; the member sizes are limited by maximum values on the diameters and thicknesses, the upper and lower ratio of the diameter to thickness, and the fact that the diameters and thicknesses must remain positive. Also, the ability to link the various design variables is included.

The method, based on the integrated approach to synthesis and analysis, places an upper constraint on the design weight and transforms the original problem into a residual minimization subject to the original constraints plus the weight constraint. This new problem, cast into the form of an unconstrained

minimization problem by using the Fiacco-McCormick technique, is solved for a series of decreasing values of the weight constraint until a near optimum design is reached.

Results for several examples were obtained. These results were compared with the results from other methods whenever possible to demonstrate the effectiveness of the algorithm.

## ACKNOWLEDGMENTS

The author wishes to express his sincerest gratitude to:

Professor Richard L. Fox, whose advice, suggestions, and encouragement was an essential element in the completion of this thesis,

Miss Diane Dryfuse for the excellent typing of this manuscript and also, for her patience,

The personnel of the Case Computing Center and Mr. Charles Kaine for their help in obtaining the numerical results for this thesis,

The National Aeronautics and Space Administration for their support under grant No. NsG 110-61.

## LIST OF SYMBOLS

<u>Symbol</u>	<u>Description</u>
$A_i$	area of the $i^{\text{th}}$ member
$C_N$	conditioning number
$C_{z1}$	a nondimensionalizing factor
$D$	an $n$ by $n$ nonsingular matrix used for scaling transformations
$D_i$	a preassigned constant for the $i^{\text{th}}$ member
$DD_j$	a set of integers
$d_i$	diameter of the $i^{\text{th}}$ member
$d_{\text{max}}$	maximum allowable diameter
$E$	Young's modulus
$g_j$	the $j^{\text{th}}$ constraint
$K_d, K_{uz}$	nondimensionalizing factors
$K_{ij}$	element of the master stiffness matrix
$\tilde{K}_{ivb}$	element of the local stiffness matrix for the $i^{\text{th}}$ member

<u>Symbol</u>	<u>Description</u>
KT	total number of loading conditions
$k_2$	cripling constant
$L_i$	length of the $i^{\text{th}}$ member
M	total number of truss members
N	total number of degrees of freedom per loading
$N_{dv}, N_{tv}$	number of diameter and thickness variables
NN	total number of nodes of the truss
nc	total number of constraints
P	load vector(s)
$P_{jk}$	load on the $j^{\text{th}}$ degree of freedom for the $k^{\text{th}}$ load condition
$PT_i$	a set of integers
R	the sum of the squares of the residual; also referred to as the residual
$r_{jk}$	residual of the $j^{\text{th}}$ degree of freedom for the $k^{\text{th}}$ load condition; sometimes called the residuals
$r_l, r_u$	lower and upper limit of the diameter to thickness ratios

<u>Symbols</u>	<u>Description</u>
$SR_{ij}$	a set of integers
$T_i$	a preassigned constant for the $i^{th}$ member
$TT_i$	a set of integers
$t_i$	thickness of the $i^{th}$ member
$t_{max}$	maximum allowable thickness
$U$	displacement vector(s)
$UU_i$	a set of integers
$u_{jk}$	displacement of the $j^{th}$ degree of freedom for the $k^{th}$ load condition
$u_{low}, u_{up}$	lower and upper limits on the displacements
$W$	weight of the truss
$W_0$	value of the weight constraint; also called the goal weight
$\bar{X}$	vector of synthesis variables
$\bar{x}^*$	the minimum of a function
$\alpha_j$	the $j^{th}$ diameter parameter
$\beta_j$	the $j^{th}$ thickness parameter

<u>Symbol</u>	<u>Description</u>
$\gamma_{ic}$	the direction cosine of the $i^{\text{th}}$ member with respect to the $c$ axis
$\epsilon_M$	tolerance on the residuals
$n\left(\frac{i}{j}\right)$	a function defined to have a value of 1 or 0; it is 1 only if $i$ equals $j$
$\lambda_{\max}, \lambda_{\min}$	maximum and minimum eigenvalues
$\nu$	the multiplier for the Fiacco McCormick formulation
$\rho$	unit weight of the material
$\sigma_b$	Euler buckling stress
$\sigma_c$	cripling stress
$\sigma_{ik}$	stress in the $i^{\text{th}}$ member for the $k^{\text{th}}$ load condition
$\sigma_{yc}, \sigma_{yt}$	the compressive and tensile yield stresses
$\phi$	the integrated analysis synthesis function

## TABLE OF CONTENTS

ABSTRACT		ii
ACKNOWLEDGMENTS		iv
LIST OF SYMBOLS		v
TABLE OF CONTENTS		ix
LIST OF FIGURES		xi
LIST OF TABLES		xii
CHAPTER I	INTRODUCTION	1
CHAPTER II	MATHEMATICAL PROGRAMMING FORMULATION OF STRUCTURAL SYNTHESIS	12
CHAPTER III	FORMULATION OF THE SOLUTION	21
A	Formulation of Method	22
B	Development of the Function $\phi$	28
CHAPTER IV	OPERATIONAL TECHNIQUES	39
A	Minimization Scheme	39
B	Scaling Transformation	44
CHAPTER V	RESULTS	52
A	Planar Case	53
1.	Three Bar Truss	53
2.	Eleven Bar Truss	55
B	Three Dimensional Cases	57
1.	Nine Bar Truss	57
2.	Twenty Five Bar Truss	59
3.	Thirty Bar Truss	61
CHAPTER VI	DISCUSSION AND CONCLUSIONS	62



FIGURES	67
TABLES	78
LIST OF REFERENCES	94
APPENDIX A	96
APPENDIX B	102

## LIST OF FIGURES

1. Three Bar Truss.
2. Design Space for Three Bar Truss.
3. An Ellipsoid in its Original (x) and Transformed (y) Coordinate Systems.
4. Value of  $\phi$  as a Function of the Number of Steps -- Rescaling Every 20 Steps.
5. Value of  $\phi$  as a Function of the Number of Steps -- Rescaling Every 7 Steps.
6. Flow Diagram for Frequency of Rescaling Logic.
7. Synthesis Path for Three Bar Truss.
8. Eleven Bar Truss, Case 2.
9. Nine Bar Truss, Case 3.
10. Twenty Five Bar Truss, Case 4.
11. Thirty Bar Truss, Case 5.

## LIST OF TABLES

1. Eigenvalues for Matrix of Second Partial Derivatives for Three Bar Truss.
2. Summary of Running Times.
3. Case 1, Displacement Limits =  $\pm 1.0$  in.
4. Load Conditions for Case 2.
5. Case 2.A, Displacement Limits =  $\pm 0.15$  in.
6. Case 2.B, Displacement Limits =  $\pm 0.25$  in.
7. Load Conditions for Case 3.
8. Case 3, Displacement Limits =  $\pm 10.0$  in.
9. Load Conditions for Case 4.
10. Case 4.A, Displacement Limits =  $\pm 0.35$  in.
11. Case 4.B, Displacement Limit =  $\pm 10.0$  in.
12. Case 4.C, Displacement Limit =  $\pm 10.0$  in.
13. Load Conditions for Case 5.
14. Case 5, Displacement Limits =  $\pm 1.60$  in.
15. Comparison of Running Times.
16. Comparison of Results.

## CHAPTER I

### INTRODUCTION

One aspect of structural design which has received considerable interest and study in recent years is the approach known as the structural synthesis process. This activity might be described as a rational process directed toward producing a design which is better than any other allowable design in achieving some quality and which satisfies a set of requirements or specifications. Typical of such qualities are the weight and cost of the structure being designed.

In the field of structural synthesis, the structural system which probably has been studied the most is the truss. A truss will be defined as a structure composed of bars which are hinged together at their ends to form a stable, rigid framework. The usual assumptions will be made considering the system:<sup>1</sup>

1. The nodes where the members are joined together are frictionless pin joints.
2. Loads and reactions are applied at the nodes.
3. The members are subjected only to axial forces (e.g. no bending moments are considered to occur).

---

<sup>1</sup>Charles Head Norris and John Benson Wilbur, Elementary Structural Analysis, Second Edition (New York: McGraw-Hill Book Company, 1960), p. 115.

Furthermore, consideration will only be extended to the linear truss. This requires that nodal displacements are small enough that the change in geometry has no significant effect on the analysis.

Since it would appear that the truss is a popular system with which to develop and study methods of structural synthesis, some of the reasons for its popularity should be listed. Among these the following might be included:<sup>1</sup>

1. It is possible to encounter a large number of degrees of freedom which is a characteristic of many interesting structural systems.
2. Methods of analysis are well developed.
3. There is a large choice of possible behavior requirements of varying complexity.

The following is a description of research into the further development of structural synthesis of the truss. In particular, it is a development within the framework of the integrated approach to engineering design as formulated by R. L. Fox and L. A. Schmit (see Reference 1).

The specific problem is to minimize the weight of a general space truss which is composed of  $M$  tubular members, joined at

---

<sup>1</sup>Richard L. Fox, "An Integrated Approach to Engineering Synthesis and Analysis", (Ph.D. Thesis, Engineering Division, Case Institute of Technology, 1965), p. 22.

NN nodes with N total nodal degrees of freedom or displacements for each load condition, and subjected to KT load conditions or configurations. The  $i$ th truss member will be described by two dimensions, namely the mean diameter  $d_i$  and wall thickness  $t_i$ . The location of the  $j$ th node is given by a set of coordinates  $(x_j, y_j, z_j)$ .

The analysis of the truss, that is, the prediction of the displacements and the stresses within the truss due to the sets of applied loads, will be accomplished by the displacement method. In general, this method can be written in the matrix form

$$[k_{ij}]U = P \quad (1.1)$$

for each load condition. The matrix  $[k_{ij}]$  is called the stiffness matrix and is computed from the geometry of the truss, the Young's modulus and cross sectional design of the members. The element  $k_{ij}$  is the force at the  $i$ th degree of freedom due to a unit displacement at the  $j$ th degree of freedom with all other possible displacements held at zero.

As an illustration of the formulation, the displacement equations for the three bar truss subjected to the two loading conditions shown in Figure 1 are

$$\left(\frac{A_1 E}{L_1}\right)U_{1x} + \left(\frac{EA_1}{2L_1} - \frac{EA_3}{2L_3}\right)U_{1y} = P_{1x}$$

$$\left(\frac{EA_1}{2L_1} - \frac{EA_3}{2L_3}\right)U_{1x} + \left(\frac{A_1 E}{2L_1} + \frac{A_2 E}{L_2} + \frac{A_3 E}{2L_3}\right)U_{1y} = P_{1y}$$

$$\begin{aligned} \left(\frac{A_1 E}{L_1}\right) U_{2x} + \left(\frac{EA_1}{2L_1} - \frac{EA_3}{2L_3}\right) U_{2y} &= P_{2x} \\ \left(\frac{EA_1}{2L_1} - \frac{EA_3}{2L_3}\right) U_{2x} + \left(\frac{A_1 E}{2L_1} + \frac{A_2 E}{L_2} + \frac{A_3 E}{2L_3}\right) U_{2y} &= P_{2y} \quad (1.2) \end{aligned}$$

where

$A_i$  is the area of member  $i$

$E$  is the Young's modulus

$L_i$  is the length of member  $i$

$P_{kj}$  is the  $j$ th component of the  $k$ th load

Unfortunately, most sets of displacement equations are not as simple to solve as the one presented above.

It has been mentioned that the final design of the truss must satisfy certain requirements or specifications. First, maintenance of structural integrity is an important requirement. To maintain structural integrity may mean that the stresses within each member be bounded by certain upper and lower limits. The material's tensile and compressive yield strengths might provide two possible limits of this type. Other limits on the stress such as the crippling stress and Euler's buckling stress could be dependent on the member's cross sectional dimensions. Second, limits are often prescribed for the displacement of the nodes. Occasionally, the function of the truss will require that limits on the displacements be specified. Also, the consideration of a linear truss

necessitates some restrictive values for the displacements. A third set of requirements may be derived from items which are not associated with the stresses and displacements of the truss. These are usually due to practical factors. For example, the ability of the production process may place a lower limit on the size to which the wall thickness can be made.

The set of conditions or constraints which will be considered here are the following:

1. All diameters  $d_i$  and thicknesses  $t_i$  will be less than the specified maximum values  $d_{\max}$  and  $t_{\max}$  respectively.
2. The diameters and thicknesses are strictly positive.
3. The ratios of member diameter to thickness  $d_i/t_i$  should remain between the given upper and lower limits  $r_u$  and  $r_\ell$ .
4. The displacements are to be within the upper and lower limits  $u_{\text{up}}$  and  $u_{\text{low}}$  respectively.
5. The member stresses are less than the material tensile yield stress and larger algebraically than the greatest of the material compressive yield stress, the Euler buckling stress, and the crippling stress.

These conditions represent some of the most common constraints placed on the design of a structure of this type.



Another requirement which will be imposed is that some of the design variables may be preassigned to have a constant value throughout the synthesis procedure. This would allow the thickness or diameter or both to be given prescribed values from the start of the design.

Furthermore, an additional requirement will be that the design of some members is made to depend directly on the design of another member by a constant ratio. This feature is called design variable linking. Linking provides the ability to require that some members of the truss be made the same and is a design practice which is often used to reduce the time or cost of fabricating the structural members and erecting the structure.

For the case of the three bar truss shown in Figure 1, the following constraints and conditions will be imposed as an example of the above discussion:

1. The stress in the  $i$ th member satisfy

$$\sigma_{yt} > \sigma_i > \max(\sigma_{yc}, \sigma_b, \sigma_c) \quad (1.3)$$

where

$\sigma_b$  is the Euler buckling stress given by

$$\sigma_b = - \frac{\pi^2 E_i (d_i^2 + t_i^2)}{8L_i^2} \quad (1.4)$$

$\sigma_c$  is the local buckling or crippling stress given by

$$\sigma_c = \frac{-k_2 E_i t_i}{d_i} \quad (1.5)$$

If  $\sigma_{yt}$  and  $\sigma_{yc}$  are 20 ksi and -15 ksi respectively, then, the constraints can be written as

$$20000 - \sigma_i > 0 \quad i = 1, 2, 3$$

$$\sigma_i + 15000 > 0 \quad i = 1, 2, 3$$

$$\sigma_i + \frac{\pi^2 E_i (d_i^2 + t_i^2)}{8L_i^2} > 0 \quad i = 1, 2, 3$$

$$\sigma_i + \frac{k_2 E_i t_i}{d_i} > 0 \quad i = 1, 2, 3$$

2. The  $j^{\text{th}}$  degree of freedom under the  $k$ -th load is limited by

$$u_{up} > u_{jk} > u_{low} \quad (1.6)$$

If  $u_{up}$  and  $u_{low}$  are  $\pm 1.0$ , the constraints would appear as

$$u_{jk} + 1.0 > 0 \quad j = x, y, \quad k = 1, 2$$

$$1.0 - u_{jk} > 0 \quad j = x, y, \quad k = 1, 2$$

3. All the thicknesses will have a value of  $1/4\pi$  inches.

4. The diameters of members 1 and 3 be linked so that

$$d_1 = d_3$$

5. The diameters are to be no larger than 10 inches and must be positive or

$$10 - d_i > 0, \quad i = 1, 2, 3 \quad (1.7)$$

$$d_i > 0, \quad i = 1, 2, 3 \quad (1.8)$$

6. The diameters will be no smaller than the thickness or

$$\frac{d_i}{t_i} - 1.0 > 0 \quad (1.9)$$

Perhaps at this point it should be emphasized that comparison of designs which satisfy the constraints is based upon the weight of the structure. The selection of weight as a criteria for comparison is common. Often the cost of the structure is assumed to be greatly dependent on the weight of the material used. This assumption appears to be an adequate way of simplifying a cost optimization problem in some cases. For application in aircraft and missiles, it is considered important to reduce the weight of the structural framework for the craft to achieve higher performances. Thus, for two designs which satisfy the constraints, the better design will be the one with the lower weight. For the case of the three bar truss of Figure 1, the weight is computed by

$$W = 7.5\rho(2.828 d_1 + d_2) \quad (1.10)$$

A number of methods for solving the minimum weight truss problem have been presented. Bibliographies listed in Reference 5

published in 1958, Reference 6 published in 1963, and Reference 7 published in 1965 give an extensive survey of the work in the general field of structural optimization which includes the truss problem. Many methods are not applicable for the general truss with the constraints considered in this paper. Some solutions fail to consider all of the constraints such as displacement limitations, others are valid for only statically determinate trusses, and a number handle only a single load condition.

For statically determinate trusses which are stress limited, the approach by F. R. Shanley, Reference 2, is quite useful. In this solution, the analysis of the truss is made and the member forces are determined. These forces are independent of the cross sectional areas of the members. The proportioning of each member is carried out for the critical force in the member so that the member does not fail due to violations of the stress constraints. If the stress constraints consider only the material strength of the material, the design simply needs to supply the member's cross sectional area. If the crippling and Euler buckling failure modes are also taken into account, then the member's cross sectional dimensions must be specified since the stress limits and dimensions are related. The basic problems associated with this solution are that it is effective only for statically determinate structures and it fails to account for constraints other than those of stress.

Another class of approaches called the method of alternate steps has received much study. In this approach, the initial

design satisfies all the conditions imposed on the truss and has a weight which is greater than the optimum weight of the truss. The weight is then reduced along the gradient of the weight function. This is called the path of steepest descent and is orthogonal to surfaces of equal weight. This reduction is carried out until a constraint is reached to within some tolerance  $\delta$ . Then a redesign, called a side step, is made on the constant weight surface in an effort to move away from the constraints. After the side step has been completed, another move along a path of steepest descent is taken until a constraint is reached once again. This cycle of steepest descent and side step is repeated until the minimum weight design is reached.

One method of alternate steps was presented by R. A. Gellatly, R. H. Gallagher, and W. A. Lubracki (see Reference 3). The size and direction of the side step is determined by the curvature of the constraints in the vicinity of the design. Another variation of the method, presented by L. A. Schmit and R. H. Mallett in Reference 4 is particularly suited to problems in which the weight surfaces are nonlinear. When a constraint is broken in the path of steepest descent and the last acceptable design is not too close to the constraints, then a steepest descent path is initiated from the last acceptable design.

It is worth noting that in moving along the path of steepest descent and in locating a suitable side step, numerous designs must be analyzed in order to check the displacement and stress

constraints. While for small systems this presents little worry, it is very time consuming for the case of a system with many degrees of freedom.

Another method is that of the integrated approach to engineering synthesis and analysis presented by R. L. Fox and L. A. Schmit, References 1 and 8. Simultaneous evolution of a design and its analysis is the basic aim of the approach. For the truss problem, the formulation requires a goal or drawdown weight  $W_0$  which is made an upper limit of a weight constraint written as

$$W_0 - W > 0 \quad (1.11)$$

A penalty function is then formed. It is a measure of how greatly the constraints are violated. A residual is also created to evaluate the degree to which the analysis is not satisfied. Finally, another function is constructed from the combination of the penalty function and the residual. By minimizing this new function, an analyzed design can be found if the goal weight is not lower than the optimum weight. As analyzed designs are found, the goal weight is reduced until it is impossible to find an analyzed design by the minimization. When this happens, the last design is accepted as an approximation to the optimum. Development of the work presented in this paper was made within the philosophy of the integrated synthesis approach.

## CHAPTER II

### MATHEMATICAL PROGRAMMING FORMULATION OF STRUCTURAL SYNTHESIS

For the purpose of this paper, structural synthesis will be defined as a rational design procedure of a structural system that, according to a definite objective, produces a design which efficiently fulfills a set of specified functions. Some of the characteristics of the methods of structural synthesis are: the absence of a priori assumptions of the failure modes for various load conditions; absence of before hand knowledge about the critical load condition for a given member; and the idea of designing the structure as an integral unit and not as a group of individual subsystems.

For the general truss as with many other structural forms, structural synthesis consists of three elements. These components are the analysis of the behavior of the design, a set of constraints, and finally, an objective or merit function which is the basis of comparison between different designs. When the elements are related in an algebraic manner, the synthesis may be considered as a mathematical programming problem which has an objective function and constraints on the behavior and design variables.

The prediction of the behavior of the structure has two main parts. They are the selection of the method of analysis and the specification of the loading conditions applied to the structure.

For the truss, several methods of analysis have been developed. The displacement, force, and force-displacement methods exemplify the variety of approaches available. Factors which have a bearing on the selection of the appropriate method include the numerical stability of the various approaches, the computer storage space required by the different methods, and the accuracy needed for the solution.

Once the selection of the method of analysis has been made, attention can be turned to the loading conditions which will be considered in the analysis. Specification of the loading conditions are commonly given in a deterministic manner. That is, the problem is usually stated so that a discrete set of loading conditions are applied to the structure to represent all of the potentially critical loadings which the structure will ever encounter. This is different from reality where loads may be changing continuously and where the magnitude of the loads have some statistical distribution.

Great care must be used in selecting the design constraints. Certainly some restrictions are obvious, however, many structural failures are due to those constraints which managed to escape the engineer's attention.

Generally constraints may be divided into two groups. The first group may be called the behavior constraints which are those restrictions that must be imposed to maintain satisfactory behavior of the structure under the applied loads. An example of



such constraints is the requirement that the truss member stresses be within the stress limits provided by the material's strength. Of course, assuring that the behavior constraints are satisfied requires that an analysis of the structure be carried out.

The second group of restrictions may be called the side constraints. These constraints include requirements on the size or proportioning of the design variables due to reasons not explicitly related to the behavior requirements. That the diameters of the truss members be limited to some maximum value may serve as an example. The motivation for such a constraint might be the desire to guarantee a sufficiently large length to diameter ratio to insure some architectural effect. The constraint given in the example is called an inequality constraint and can be written as either

$$d_{\max} > d_i \quad (2.1a)$$

or

$$d_{\max} - d_i > 0 \quad (2.1b)$$

Linking of the design variables may also be considered a side constraint. This might appear as an equality constraint. That is, member  $i$  of a truss could be made to have the same diameter as member  $j$ . This would be written as

$$d_i = d_j \quad (2.2)$$

Such a requirement is probably due to some external reason such as the cost or ease of construction.

It should be noted that constraints may be either linear or nonlinear. An example of a linear constraint is the requirement that the diameter of a truss member be less than some maximum value. An example of a nonlinear constraint is that the Euler buckling stress be less than the stress in the member or

$$\sigma - \sigma_b > 0$$

where, for a tubular member,

$$\sigma_b = - \frac{\pi^2 E (d^2 + t^2)}{8L^2}$$

$\sigma$  is the stress in the member.

Generally, it is harder to work with the nonlinear constraints and, unfortunately, nonlinear constraints seem to occur more often than linear constraints in real structural problems.

The objective function is the basis of distinction between different designs which satisfy the constraints. Common ones include the weight or cost of the structure. Other possible objective functions are aerodynamic performance or heat transfer rate.

The importance of the selection of the loading system and constraints should be emphasized. The omitting of a critical load condition or neglecting an important failure mode is the usual cause of structural failures. No matter how efficient the

synthesis procedure is, if an active load condition is not recognized, or if a significant failure mode is overlooked, the final design is apt to have unsatisfactory behavior if not collapse.

Once the analysis, constraints, and objective function have been specified, the formulation of structural synthesis as a mathematical programming problem may be accomplished.

Basic to the formulation is a set of quantities which are allowed to vary throughout the synthesis. These quantities are called design variables. Those quantities which do not vary and are given values at the initiation of the process are called preassigned parameters. When each of the design variables are assigned an axis in an n-dimensional Cartesian space, the space will be known as a design variable space and a given set of values of the design variables are represented by a point in that space.

In the design variable space, all points at which there is incipient failure to satisfy a particular requirement form a constraint surface. If the design is slightly on one side of this surface, the design fails to satisfy that particular constraint, while if slightly on the other side of the surface, the design does satisfy that constraint. If the constraint surface is determined by behavioral constraints, then the surface is called a behavior constraint surface. Otherwise, it may be said to be a side constraint surface. In structural problems there is a behavior constraint surface for each load condition for a particular failure mode. If there are more than one requirement on the design,

which is the most common case, then the collection of all constraint surfaces is called the composite constraint surface.

Designs which are within the composite constraint surface, that is, designs that fulfill all the requirements, are deemed acceptable designs. If even one constraint is not satisfied by a design, it is called an unacceptable design. A design which is on the constraint surface (or at least within a region  $\epsilon$  from the surface) is called a bound point and the surface is considered to be an active constraint. If a point is not a bound point, then it is a free point.

With the above description, the general truss problem presented in Chapter I may be stated as follows:

Given: preassigned parameters, the loading system with one or more distinct load conditions, and behavior and side constraints;

Find: the set of design variables subject to the side and behavior constraints such that the weight  $W$  is given a minimum value.

As an example of some of the terms mentioned in the previous discussion, the design space of the three bar truss problem discussed in Chapter I will be described. In Figure 2, the coordinate axes are labeled for the member diameters  $d_1$  and  $d_2$ . Note that the area of member  $i$  is given by

$$A_i = \pi t_i d_i \quad (2.3)$$

Since the presentation of the example in Chapter I required

$$t_i = 1/4\pi$$

then

$$A_i = d_i/4 \quad (2.4)$$

Curves 1, 2, and 3 correspond to the compressive and tensile stress constraints. In particular, Curve 1 is associated with

$$\sigma_{11}(d_1, d_2) = \sigma_{32}(d_1, d_2) = -15000 \quad (2.5a)$$

Curve 2 is associated with

$$\sigma_{12}(d_1, d_2) = \sigma_{21}(d_1, d_2) = 20000 \quad (2.5b)$$

and Curve 3 is associated with

$$\sigma_{21}(d_1, d_2) = \sigma_{22}(d_1, d_2) = 20000 \quad (2.5c)$$

where

$\sigma_{jk}$  indicates the stress in the  $j^{\text{th}}$  member for the  $k^{\text{th}}$  load condition.

There is an additional constraint that the diameters of the members must be positive. This makes the  $d_1$  and  $d_2$  axes constraints also. The requirement that the diameters are positive is a side

constraint; the demands placed on the stresses are behavior constraints. All of the constraints form the composite constraint surface. The various points A, B, C, D, and E of Figure 2 can be described in the following manner:

- A - free, acceptable
- B - free, unacceptable
- C - bound, acceptable
- D - bound, unacceptable
- E - optimum design

Curve 4 represents a weight contour. Specifically, it is a line of designs which all weigh 150 pounds if  $\rho$  is equal to unity. Curve 5 is another weight contour for designs weighing 100 pounds. Indeed, any line parallel to Curves 4 or 5 will be a weight contour. The weight for this problem is an example of a linear objective function and can be expressed in the form

$$W = \sum_{i=1}^3 k_i D_i \quad (2.6)$$

where

$D_i$  is the diameter of the member  $i$

$k_i$  is a constant.

The assertion that nonlinear constraints are more difficult to work with than linear constraints has been made. For the

problem with linear constraints, one method of solution is to move from one intersection of constraints to another in such a way so as to reduce the objective function. Since, for linear objective functions, an optimum is at one of these intersections, the approach will converge to the solution. For problems with nonlinear constraints, movement from intersection to intersection is quite difficult. Furthermore, there is no guarantee that the solution is at an intersection.

Due to the difficulties with the nonlinear constraints, various approaches for optimization such as the method of alternate steps have been developed. Regrettably, the use of these approaches necessitates numerous analyses of the designs along the synthesis path. To avoid this, the integrated approach to synthesis and analysis was developed. The formulation of a method within the philosophy of the integrated approach is presented in the next chapter.

### CHAPTER III

#### FORMULATION OF THE SOLUTION

The problem of optimizing the weight of a general truss has been presented in Chapter I with a description of the constraints and the method for analysis. To illustrate the different aspects of the proposed synthesis, an example of a three bar truss with various constraints was given. A more concise statement of the problem is

Given: preassigned parameters, the loading system with one or more distinct load conditions, and behavior and side constraints;

Find: the set of design variables subject to the side and behavior constraints such that the weight  $W$  is given a minimum value.

Some of the difficulties associated with the many approaches to the problem were mentioned in the previous chapters. For those approaches which are able to handle a variety of load conditions and constraints, the solutions require numerous analyses. This difficulty is emphasized whenever the system has a large number of degrees of freedom. In an effort to avoid this problem, the integrated approach to synthesis attempts to perform the design and analysis simultaneously. Thus, the synthesis variables include both the behavior and member design variables. The behavior variables consist of the displacements of the nodes, stresses



within the members, or some other set of quantities which can fully describe the behavior of the structure under all the loading conditions. The member design variables are those parameters which specify the cross sectional dimensions of the truss members. Attention should be drawn to the fact that the number of behavior variables is dependent on the method of analysis.

#### A. Formulation of Method

Since the integrated approach attempts to evolve designs and their analyses concurrently, it would seem desirable that the new design have a weight which is less than the weight of the previously analyzed designs. One method to accomplish this is presented in the development of the integrated approach by R. L. Fox in Reference 1. The method is to select a goal weight  $W_0$  which is made the upper limit of a constraint on the weight written as

$$g_w = W_0 - W \geq 0 \quad (3.1)$$

Once this is done, the weight is treated as any other constrained parameter and any analyzed design which satisfies all the constraints will be at most equal to the goal weight. If the goal weight is selected so that it is progressively reduced, then the weights of the designs will be successively smaller.

After the weight constraint has been imposed, the primary consideration is the development of a method to analyze a design and a procedure to keep within the constraints. These involve

the construction of a measure of the degree to which the analysis is satisfied and selection of a standard mathematical programming technique.

The displacement method of analysis given by Equation (1.1) can be written as

$$f_i(\bar{x}, \bar{D}) = 0 \quad , \quad i = 1, 2, \dots, n \quad (3.2)$$

where  $\bar{x}$  is a unique behavior for the design variables  $\bar{D}$ . Since such a form does exist, then one method of finding the solution  $\bar{x}_j$  for a given design  $\bar{D}_j$  is

1. Define a function called the residual as

$$R(\bar{x}) = \sum_{i=1}^n [f_i(\bar{x}, \bar{D})]^2 \quad (3.3)$$

2. Find  $\bar{x}$  such that

$$R(\bar{x}) \rightarrow \text{MIN} = 0$$

The minimization of  $R(\bar{x})$  can be accomplished by using one of the minimization schemes developed for an unconstrained function. Two possible schemes, the Fletcher-Powell and the Fletcher-Reeves methods, will be described in another chapter.

In the synthesis process, both the behavior variables  $\bar{x}$  and the design variables  $\bar{D}$  are allowed to vary at the same time. If

$$\bar{X} = \begin{pmatrix} \bar{x} \\ \bar{D} \end{pmatrix}$$

then the problem of finding  $\bar{X}$  such that  $R(\bar{X})$  is reduced to zero has no unique solution. For the displacement method of analysis, the residual is defined as

$$R(\bar{X}) = \sum_{k=1}^{KT} \sum_{i=1}^N r_{ik}^2 \quad (3.4)$$

where

$$[r_{ik}] = [K_{ij}][u_{jk}] - [P_{ik}] \quad (3.5)$$

The element  $r_{ik}$  will be called the residual of the  $i^{\text{th}}$  degree of freedom for the  $k^{\text{th}}$  load condition or simply a residual. If there is possible confusion due to the common usage of the word "residual", the meaning will be made clear by writing the appropriate symbol for the quantity in question.

Another consideration is the criteria for which the equations of analysis will be considered satisfied. One possible criterion can be written as

$$R \leq \epsilon_R$$

This requirement is not particularly adequate since it does not provide control over the individual residuals  $r_{ik}$ . That is, if all the residuals are zero except for one, say the  $pq^{\text{th}}$  element, then the analysis might be accepted even though the  $pq^{\text{th}}$  residual is quite large. To avoid this, a criterion which provides direct control on the magnitude of the individual residuals was adopted.

The condition is of the form

$$|r_{ik}| \leq r_M \quad \begin{matrix} i = 1, 2, \dots, N \\ k = 1, 2, \dots, KT \end{matrix} \quad (3.6)$$

In the statement of the problem there is a set of behavior and side constraints. Besides these constraints, there is the additional constraint imposed on the weight by Equation (3.1). The convention that the inequality constraints be written so as to be positive when the restrictions are satisfied will be adapted. For example,

$$g_k = d_{\max} - d_i \geq 0$$

is a constraint on the maximum size of the diameter of the  $i^{\text{th}}$  member.

At this point the problem may be restated as

Given: preassigned parameters, behavior and side constraints, the loading system with one or more distinct loading conditions, and the weight constraint;

Find: the set of behavior and design variables subject to the behavior, side, and weight constraints such that the residual  $R$  is given a minimum value.

If at the minimum of this problem the analysis cannot be satisfied,

then the goal weight is less than the optimum weight.

The minimization problem presented above, which is merely a transformation of the original problem, can be cast into the form of a Fiacco-McCormick function. That is, construct the function

$$\phi(X, v) = R(X) + v \left[ \sum_{i=1}^{nc} \frac{a_i}{g_i(X)} + \frac{a_w}{g_w(X)} \right] \quad (3.7)$$

where  $(v > 0)$  and  $a_i$  and  $a_w$  are constants. Choose an initial multiplier  $v > 0$  and an acceptable point  $X_0$  which is strictly within the constraint set. If  $X$  should fall outside of the constraint set, then let the function have a value

$$\phi(X, v) \rightarrow +\infty$$

Intuitively, it would seem likely that the minimum would always fall within the constraint set. Thus, the problem is one of minimizing  $\phi$  starting at  $X_0$  using an unconstrained minimization scheme. Let the minimum of  $\phi(X, v_i)$  be a point denoted by  $X(v_i)$ .

Once  $X(v_1)$  is determined, the process is repeated for a sequence of  $v$  values such that  $v_1 > v_2 > \dots > v_j > 0$ . Each minimum  $X(v_i)$  will be strictly within the constraint set. Also, with each reduction in  $v$  the value of

$$v \left[ \sum_{i=1}^{nc} \frac{a_i}{g_i} + \frac{a_w}{g_w} \right] \quad (3.8)$$

is reduced in influence which, in turn, forces the reduction of

$R(X)$  in the minimization of  $\phi$ . As  $v$  is reduced, the design  $X(v_i)$  is allowed to approach closer and closer to the active constraints if such a move will be beneficial in reducing  $\phi$ . Eventually, if a goal weight is equal to or greater than the true optimum weight for the problem, the residual term should be forced to zero. The process is normally not carried out that far since the analysis is considered satisfied when the condition given by Equation (3.6) is met. If it becomes apparent that the residual  $R(X)$  cannot be reduced enough to meet the requirements of Equation (3.6) with further reduction of  $v$ , then the process is terminated and the last design satisfying the analysis is considered to be an approximation of the optimum design.

If for a particular goal weight  $W_{0,i}$  an analyzed design with a weight of  $W_i$  can be found, then a new goal weight is selected. A method which might be used to select this goal weight is given by the formula

$$W_{0,i+1} = (1.0 - \Delta)W_i \quad (3.9)$$

where  $\Delta$  is fairly small. That is, the goal weight is set equal to some percentage of the weight of the last analyzed design. The process could be started with a fairly coarse value of  $\Delta$ , say 0.10. When a goal is selected for which the analysis cannot be satisfied, then a smaller value of  $\Delta$  could be used and the process started once again from the last analyzed design. The value of  $\Delta$  could thus be refined until the reduction in weight would

not be worth the effort to achieving it.

#### B. Development of the Function $\phi$

To implement the method of synthesis outlined above, the function  $\phi$  must be developed explicitly. To facilitate the formulation, several sets or tables of integers were constructed. Since some use will be made of these in developing  $\phi$ , they will be defined as follows:

1. The  $PT_{ij}$  table has dimension of  $M$  by  $2$  where  $M$  is the number of truss members. The node numbers of the two joints which are joined by the  $i$ th member are given by  $PT_{i1}$  and  $PT_{i2}$ .
2. The  $UU_j$  table has a single dimension of  $3NN$  where  $NN$  is the total number of nodes. For a given  $i$ th node, the displacement indices for the  $x$ ,  $y$ , and  $z$  spacial directions are given by the value of  $UU_{3i-2}$ ,  $UU_{3i-1}$ , and  $UU_{3i}$  respectively. If the node cannot have a displacement in a given direction  $k$  due to boundary conditions (i.e. the direction is supported by some external restraint), then the value of  $UU_k$  is zero.
3. The  $SR_{ij}$  table has dimensions  $M$  by  $6$ . It is formed by the combination of the  $PT_{ik}$  and the  $UU_j$  table. The indices of the displacements in the  $x$ ,  $y$ , and  $z$  spacial directions of one of the two nodes which are

joined together by the  $i^{\text{th}}$  member are given by the values of  $SR_{i1}$ ,  $SR_{i2}$ ,  $SR_{i3}$  respectively. The indices of the displacement of the other node in the  $x$ ,  $y$ , and  $z$  directions are similarly given by the values  $SR_{i4}$ ,  $SR_{i5}$ , and  $SR_{i6}$ . Again, if a displacement in a given direction  $k$  for one of the nodes of some member  $i$  is forced to zero by an external restraint, then the value of  $SR_{ik}$  is equal to zero.

4. The  $DD_i$  table has dimension  $M$  and is used to coordinate the diameter variables. For some member  $i$ , the index  $j$  of the design variable  $\alpha_j$  which describes the diameter of the  $i^{\text{th}}$  member is given by  $DD_i$ . If the diameter is fixed for the  $i^{\text{th}}$  member, then  $DD_i$  is given the value of zero.
5. The  $TT_i$  table has a dimension  $M$  and performs the same function for the truss members' thicknesses as the  $DD_i$  does for the members' diameters by supplying the index for the thickness design variables  $\beta_j$ .

After the tables have been clarified, the formulas for several useful quantities may be given. The length of the  $i^{\text{th}}$  member which is connected to the nodes  $p$  and  $q$  may be written as

$$L_i = [(x_p - x_q)^2 + (y_p - y_q)^2 + (z_p - z_q)^2]^{1/2} \quad (3.10)$$



where  $(x_j, y_j, z_j)$  is the coordinate of the  $j^{\text{th}}$  node.

The direction cosines of the  $i^{\text{th}}$  member, again with nodes  $p$  and  $q$ , to the  $x$ ,  $y$ , and  $z$  coordinate axes are respectively given by

$$\begin{aligned} r_{ix} &= (x_q - x_p)/L_i \\ r_{iy} &= (y_q - y_p)/L_i \\ r_{iz} &= (z_q - z_p)/L_i \end{aligned} \quad (3.11)$$

The mean diameter  $d_i$  of the  $i^{\text{th}}$  member is given by

$$d_i = D_i \alpha_{DD_i} \quad (3.12)$$

where  $D_i$  is a preassigned constant such that it is either a prescribed diameter or a prescribed multiplier for the diameter parameter  $\alpha_{DD_i}$  and is fixed throughout the entire synthesis. If it is the former possibility, then by the definition of the tables,

$$DD_i = 0$$

Thus, to obtain

$$d_i = D_i$$

requires that

$$\alpha_0 = 1.0 \quad (3.13)$$

Likewise, the thickness  $t_i$  will be given by

$$t_i = T_i \beta_{TTi} \quad (3.14)$$

which functions in the same manner as the computation of the diameter.

The equilibrium equations used for the analysis of the behavior of the truss is given by

$$\sum_{j=1}^N K_{xj} u_{jk} = P_{xk} \quad \begin{aligned} x &= 1, 2, \dots, N \\ k &= 1, 2, \dots, KT \end{aligned} \quad (3.15)$$

where

$N$  is the number of degrees of freedom per load condition

$KT$  is the total number of load conditions

$K_{xj}$  is the force at the  $x^{\text{th}}$  degree of freedom due to a unit displacement in the  $j^{\text{th}}$  node. It is called an element of the master stiffness matrix.

One method for computing  $K_{xj}$  is from the element stiffness matrix denoted by  $[K_{vb}]_i$  or  $[K_{ivb}]$  for each member  $i$  of the truss. That is, an element of the master stiffness matrix may be found by the formula

$$K_{xj} = \sum_{i=1}^M \sum_{v=1}^6 \sum_{b=1}^6 \bar{K}_{ivb} \left[ r_i \left( \frac{SR_{iv}}{l} \right) r_j \left( \frac{SR_{jb}}{l} \right) \right] \quad (3.16a)$$

where

$$n\left(\frac{i}{j}\right) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad (3.16b)$$

This is an inefficient way of assembling the master stiffness matrix if each element is formed one by one. Generally, if such a matrix is formed, it is formed at one time by considering all combinations of the indices  $(SR_{iv}, SR_{ib})$  for each individual member. If this is done, then the method becomes very useful. In the present problem, the master stiffness matrix was never formed to conserve computer storage space.

If the direction cosines for the  $i^{\text{th}}$  member with respect to the standard coordinate system  $x, y$ , and  $z$  are  $\gamma_{ix}$ ,  $\gamma_{iy}$ , and  $\gamma_{iz}$  respectively, then the matrix  $QL_i$  shall be defined as

$$QL_i = \begin{bmatrix} \gamma_{ix}^2 & \gamma_{ix}\gamma_{iy} & \gamma_{ix}\gamma_{iz} \\ \gamma_{iy}\gamma_{ix} & \gamma_{iy}^2 & \gamma_{iy}\gamma_{iz} \\ \gamma_{iz}\gamma_{ix} & \gamma_{iz}\gamma_{iy} & \gamma_{iz}^2 \end{bmatrix} \quad (3.17)$$

Now the local stiffness matrix in the standard coordinate system may be written in the form of the partitioned matrix

$$[K_{ivb}] = \frac{A_i E_i}{L_i} \begin{bmatrix} QL_i & -QL_i \\ -QL_i & QL_i \end{bmatrix} \quad (3.18)$$

where

$A_i$  is the area of the member or  $\pi d_i t_i$

$E_i$  is Young's modulus for the member

In forming the value of the residual  $R$  as described in Chapter 3, the residuals  $r_{\ell k}$  are first computed. This is written as

$$r_{\ell k} = \sum_{j=1}^n K_{\ell j} u_{jk} - P_{\ell k} \quad \begin{array}{l} \ell = 1, 2, \dots, N \\ k = 1, 2, \dots, KT \end{array} \quad (3.19)$$

Now  $R$  can be given by

$$R = \sum_{k=1}^{KT} \sum_{\ell=1}^N r_{\ell k}^2$$

If a local force element for member  $i$  is denoted by  $\tilde{r}_{vk}^i$ , then

$$\tilde{r}_{vk}^i = \sum_{b=1}^6 [\tilde{K}_{ivb} u_{SR_{ib},k}] \quad (3.20)$$

The residual  $r_{\ell k}$  for a given load condition  $k$  and degree of freedom  $\ell$  is found by

$$r_{\ell k} = \sum_{i=1}^M \sum_{v=1}^6 \left[ \tilde{r}_{vk}^i n \left( \frac{SR_{iv}}{\ell} \right) \right] - P_{\ell k} \quad (3.21)$$

Again it is unlikely that each of the residuals would be computed one by one, but rather, like the master stiffness matrix, all residuals resulting from one load condition would be formed at

one time. By noting the form of the partitioned matrix in Equation (3.18), considerable computational efforts can be saved in forming the residuals. The above formulation of  $R$  completes part of the function  $\phi$  as given by

$$\phi = R + v \left[ \sum_{i=1}^{nc} \frac{a_i}{g_i} + \frac{a_w}{g_w} \right]$$

In the next part, attention is turned to the constraints  $g_i$  and the weight constraint  $g_w$ .

The first constraints to be considered will be those of stress. The computation of the stress in the  $i^{\text{th}}$  member with nodes  $p$  and  $q$  from the  $k^{\text{th}}$  loading is given by

$$\begin{aligned} \sigma_{ik} = \frac{E_i}{L_i} & \left| \gamma_{ix}(u_{px,k} - u_{qx,k}) + \gamma_{iy}(u_{py,k} - u_{qy,k}) \right. \\ & \left. + \gamma_{iz}(u_{pz,k} - u_{qz,k}) \right| \end{aligned} \quad (3.22)$$

where  $u_{jc,k}$  is the displacement at the  $j^{\text{th}}$  node in the direction of the  $c$  axis for the  $k^{\text{th}}$  load condition. The values of the crippling and Euler buckling of a thin wall tubular member as a function of the member's mean diameter  $d_i$  and thickness  $t_i$  are respectively given by

$$\begin{aligned} (\sigma_{cr})_i &= - \frac{k_2 E_i t_i}{d_i} \\ (\sigma_b)_i &= - \frac{\pi^2 E_i (d_i^2 + t_i^2)}{8 L_i^2} \end{aligned} \quad (3.23)$$

where  $k_2$  is a material constant.

Listing the constraints which must be satisfied in each member for every load condition gives the constraints, as specified earlier,

$$\begin{aligned}
 g_1 &= \sigma_{yt} - \sigma_{ik} > 0 \\
 g_2 &= \sigma_{ik} - \sigma_{yc} > 0 \\
 g_3 &= \sigma_{ik} - (\sigma_{cr})_i > 0 \\
 g_4 &= \sigma_{ik} - (\sigma_b)_i > 0
 \end{aligned} \tag{3.24}$$

Without further ado, the contribution to  $\phi$  due to the stress constraints is

$$\phi_\sigma = \sum_{k=1}^{KT} \sum_{i=1}^M K_d \left[ \frac{1.0}{\sigma_{yt} - \sigma_{ik}} + \frac{1.0}{\sigma_{ik} - \sigma_{yc}} + \frac{1.0}{\sigma_{ik} - (\sigma_{cr})_i} + \frac{1.0}{C_{z1}(\sigma_{ik} - (\sigma_b)_i)} \right] > 0 \tag{3.25}$$

where  $K_d$  and  $C_{z1}$  are nondimensionalizing factors.

The constraints which were placed on the displacements in the specification of the problem were

$$\begin{aligned}
 g_5 &= u_{up} - u_{jk} > 0 \\
 g_6 &= u_{jk} - u_{low} > 0
 \end{aligned} \tag{3.26}$$

where  $u_{up}$  and  $u_{low}$  are the upper and lower limits on the

displacements. The effect of the function due to these constraints is given by

$$\phi_d = \sum_{k=1}^{KT} \sum_{j=1}^N K_{uz} \left[ \frac{1.0}{u_{up} - u_{jk}} + \frac{1.0}{u_{jk} - u_{low}} \right] \quad (3.27)$$

where  $K_{uz}$  is a nondimensionalizing constant.

Side constraints in this problem have been imposed on the truss members. These constraints may be written as

$$\begin{aligned} g_7 &= d_{max} - d_i > 0 \\ g_8 &= t_{max} - t_i > 0 \\ g_9 &= d_i > 0 \\ g_{10} &= r_u - d_i/t_i > 0 \\ g_{11} &= d_i/t_i - r_\ell > 0 \end{aligned} \quad (3.28)$$

The total effect on  $\phi$  from all of these constraints is

$$\begin{aligned} \phi_s = \sum_{i=1}^M \left[ \frac{1.0}{d_{max} - d_i} + \frac{1.0}{t_{max} - t_i} + \frac{1.0}{d_i} + \frac{1.0}{r_u - d_i/t_i} \right. \\ \left. + \frac{1.0}{d_i/t_i - r_\ell} \right] \end{aligned} \quad (3.29)$$

Finally, consideration must be given to the weight constraint. The weight is given by

$$W = \sum_{i=1}^M w f_i^{\alpha} D D_i^{\beta} T T_i \quad (3.30)$$

where

$$w\bar{r}_i = \pi \rho D_i T_i$$

$\rho$  = unit weight of material

Since the constraint may be written as

$$g_W = W_0 - W > 0$$

then the contribution to the function  $\phi$  is

$$\phi_W = \frac{W_m W_0}{W_0 - W} \quad (3.31)$$

where

$W_0$  is used as the goal weight and as a non-dimensionalizing factor.

$W_m$  is used to vary the influence of  $\phi_W$  with respect to  $\phi$ .

Once all of the constraints are considered, the function  $\phi$  may be written first as

$$\phi = R + v(\phi_\sigma + \phi_d + \phi_s + \phi_W) \quad (3.32)$$

and then in its full form



$$\begin{aligned}
\phi(X) = & \sum_{k=1}^{KT} \sum_{\ell=1}^N r_{\ell k}^2 \\
& + v \left\{ K_d \sum_{k=1}^{KT} \sum_{i=1}^M \left[ \frac{1.0}{\sigma_{yt} - \sigma_{ik}} + \frac{1.0}{\sigma_{ik} - \sigma_{yc}} \right. \right. \\
& \quad \left. \left. + \frac{1.0}{c_{z1}(\sigma_{ik} - (\sigma_b)_i)} + \frac{1.0}{\sigma_{ik} - (\sigma_{cr})_i} \right] \right. \\
& + K_{uz} \sum_{k=1}^{KT} \sum_{j=1}^N \left[ \frac{1.0}{u_{up} - u_{ij}} + \frac{1.0}{u_{ij} - u_{low}} \right] \\
& + \sum_{i=1}^M \left[ \frac{1.0}{r_u - d_i/t_i} + \frac{1.0}{d_i/t_i - r_\ell} + \frac{1.0}{d_i} \right. \\
& \quad \left. + \frac{1.0}{d_{max} - d_i} + \frac{1.0}{t_{max} - t_i} \right] \\
& \left. + \frac{W_o W_m}{W_o - W} \right\} \tag{3.33}
\end{aligned}$$

The gradient of the function will be needed in the minimization process. With the function in hand, the gradient is found by some tedious but straight forward operations. The results of such operations are presented in Appendix A.

## CHAPTER IV

### OPERATIONAL TECHNIQUES

In order to handle realistic problems, the approach to synthesis outlined in the previous chapter was programmed for the computer. To make the formulation operational, certain procedures or devices were incorporated in the final program. Those which are discussed in this chapter are the minimization scheme and a scaling transformation.

#### A. Minimization Scheme

Central to this method of synthesis is the minimization of the function  $\phi$ . Since the function is parametrically dependent on the values of  $v$  and the goal weight  $W_0$ , a new minimization is started every time  $v$  or  $W_0$  is modified. Occasionally, two or three minimizations are required to locate one analyzed design for a given goal weight.

The possible techniques used for these minimizations are the same as those associated with unconstrained minimization problems. Presented here are two important methods of this type. They are the Fletcher-Reeves method of conjugate gradients and the Fletcher-Powell method (see References 11 and 12). The relative merits and drawbacks of each will be discussed and the reasons for the selection of the Fletcher-Reeves method to obtain numerical results for the test problems will be given. Before the descriptions of the methods mentioned above are presented, some

of the considerations which should be made in selecting a minimization procedure will be set forth.

Due to the fact that there are a number of minimizations performed in this synthesis approach, it seems natural that an efficient minimization scheme would be desirable. By efficient, it is meant that the scheme should be able to find the minimum of the function in a relatively small number of operations. This need is even more strongly emphasized since practical problems have a large number of variables. Even for efficient schemes, the amount of work is usually dependent in some degree on the number of variables in the problem. A second important consideration is the ability to handle a large number of variables in terms of computer storage. For a moderately complex truss system with several load conditions the number of variables could become rather large. After all factors have been considered, it may become obvious that the selection of a method will require compromises in the goals mentioned above.

The minimization of a continuous function  $f$  with continuous first and second partial derivatives will be considered. The function  $\phi$  satisfies such conditions except where the constraints are exactly critical. Since the region where the minimization search is taking place is strictly within the constraints the methods which can be used for minimizing  $f$  are also applicable for  $\phi$ .

From the standard quadratic expansion about the point  $\bar{x}^0$

given by

$$f = f^0 + \sum_{i=1}^n a_i (x_i - x_i^0) + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n G_{ij} (x_i - x_i^0) (x_j - x_j^0) \quad (4.1)$$

where

$$G_{ij} = \left( \frac{\partial^2 f}{\partial x_i \partial x_j} \right)_{\bar{x}^0}$$

$$a_j = \left( \frac{\partial f}{\partial x_j} \right)_{\bar{x}^0}$$

the vector between the point  $\bar{x}$  and the minimum  $\bar{x}^*$  can be shown to be

$$\bar{x}^* - \bar{x} = -G^{-1} \nabla f(\bar{x}) \quad (4.2)$$

To approximate the matrix  $G^{-1}$  in Equation (4.2), an  $n$  by  $n$  matrix  $H$  is employed. Initially chosen to be positive definite,  $H$  is modified in a manner so that it tends to  $G^{-1}$  as the minimization process is accomplished. The algorithm for the Fletcher-Powell method is

Select  $H_0$  as any arbitrary positive definite matrix;  
usually the identity matrix.

$\bar{x}_0$  as an initial point.

then

$$\bar{s}_i = -H_i \nabla f(\bar{x}_i)$$

$$f_i = \text{minimum of } f(\bar{x}_i + \alpha_i \bar{s}_i) \text{ with respect to} \\ \text{the variable } \alpha_i \text{ occurring at } \alpha_i^*$$

$$\bar{\sigma}_i = \alpha_i^* \bar{s}_i$$

$$\bar{x}_{i+1} = \bar{x}_i + \bar{\sigma}_i$$

$$H_{i+1} = H_i + A_i + B_i$$

where

$$A_i = \frac{\bar{\sigma}_i \bar{\sigma}_i^T}{\bar{\sigma}_i^T \bar{y}_i}$$

$$B_i = \frac{-H_i \bar{y}_i \bar{y}_i^T H_i}{\bar{y}_i^T H_i \bar{y}_i}$$

$$\bar{y}_i = \nabla f(\bar{x}_{i+1}) - \nabla f(\bar{x}_i)$$

Fletcher and Powell showed that if certain precautions are taken,  $H$  will always remain positive definite. This guarantees that the method will converge since the function  $f$  initially decreases along the direction  $\bar{s}_i$  causing the function to be decreased for each iteration of the process. Also, for a quadratic function of  $n$  variables, it was shown that after  $n$  or fewer steps, the minimum will be reached and  $H$  will converge to the inverse of the matrix of second partial derivatives.

The Fletcher-Reeves method of conjugate gradients is one in which a direction for minimization  $\bar{s}_i$  is generated from the gradient and the previous direction  $\bar{s}_{i-1}$ . Specifically, the algorithm for the routine is as follows:

Choose  $\bar{x}_0$  = an arbitrary initial point

with  $\bar{s}_0 = \nabla f(\bar{x}_0)$

then

$$\bar{x}_{i+1} = \bar{x}_i + h^* \bar{s}_i \text{ where } h^* \text{ is such that}$$

$$f(h) = f(\bar{x}_i + h \bar{s}_i) \text{ is minimized at}$$

$$h = h^* .$$

$$\bar{g}_{i+1} = \nabla f(\bar{x}_{i+1})$$

$$\beta_i = \bar{g}_{i+1}^2 / \bar{g}_i^2$$

$$\bar{s}_{i+1} = \bar{g}_{i+1} + \beta_i \bar{s}_i$$

The Fletcher-Reeves method is also guaranteed, except for effects due to rounding errors, to find a minimum of any quadratic function of  $n$  variables in  $n$  or fewer steps.

Even though for the quadratic function both methods can theoretically achieve a minimum in the same number of steps, the Fletcher-Powell method is generally more powerful. For ill behaved functions, seemingly the natural condition for real problems, the Fletcher-Powell method does produce a minimum more

efficiently than the Fletcher-Reeves method. Unfortunately, the Fletcher-Powell method requires the storage and manipulation of an  $n$  by  $n$  matrix. This is undesirable since the storage of very large matrices requires highly time consuming operations such as reading and writing on drums and tapes. The Fletcher-Reeves method, however, requires the storage of only a few  $n$  vectors.

Despite the fact that the Fletcher-Reeves method is less powerful, it was chosen as the minimization scheme to use in the study of this synthesis approach. The loss of efficiency in minimization was felt to be off set by an increase in capability for handling large problems.

#### B. Scaling Transformation

In the first tests of this synthesis approach, it was found that the function  $\phi$  was often difficult to minimize using the Fletcher-Reeves method. After three or four steps, the process would produce very little reduction of the function value and small changes in the design vector. Even after  $3n/2$  or  $2n$  steps, where  $n$  was the total number of synthesis variables, an insignificant amount of progress would be made. From the elements of the gradient, it was clear that the design vector was not at or even near the minimum. In essence, many of the characteristics of an ill conditioned function were present.

In an effort to improve the conditioning of the function, a

scaling transformation was considered. Such an operation is a multiplication of the variables by constants and can be represented by the matrix formulas

$$Y = D^{-1}X \quad (4.3a)$$

$$X = DY \quad (4.3b)$$

where

$X$  is the  $n$  vector in the original system.

$Y$  is the  $n$  vector in the transformed system.

$D$  is an  $n$  by  $n$  nonsingular diagonal matrix

which contains the multiplying constants.

The ability of such a transformation to improve the conditioning of a function can be explained in terms of eigenvalues and the Gerschgorin Theorem.

#### Gerschgorin's Theorem

Every eigenvalue of the matrix  $A$  lies in at least one of the disks centered at  $a_{ii}$  and of radii

$$R_i \equiv \sum_{j \neq i} |a_{ij}| \quad (4.4)$$

It will become clear that a characteristic of an effective transformation is its ability to reduce the range in which the eigenvalues of the matrix  $A$  lie.



Consider the quadratic function  $f$  given by

$$f = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} \quad (4.5a)$$

where

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = a_{ij} \quad (4.5b)$$

is the form of its second partial derivatives. If  $f$  is assigned a value and if  $A$  is positive definite, then the locus of points generated by Equation (4.5a) is an ellipsoidal hypersurface. The ratio of the principal axes are proportional to the square root of the ratio of the eigenvalues obtained from  $A$  (i.e. the matrix of second partial derivatives). If all the eigenvalues are equal, the hypersurface is a sphere. For this special case of all the eigenvalues of  $A$  being equal, the function is very easy to minimize. Indeed, the conditioning of a function for minimization is usually related to the conditioning number  $C_N$  defined as the ratio of the largest to least eigenvalue or

$$C_N = \lambda_{\max} / \lambda_{\min} \quad (4.6)$$

and the least value of  $C_N$  is 1.0 which occurs when all the eigenvalues are equal.

The function  $f$  of Equation (4.5) can be written in terms of the transformed variables as

$$f = \frac{1}{2} Y^T \tilde{A} Y \quad (4.7)$$

where

$$\tilde{A} = D^T A D \quad (4.8)$$

One selection for the transformation constants may be given by

$$\begin{aligned} d_{ii} &= 1/(c\sqrt{a_{ii}}) \\ d_{ij} &= 0 \quad i \neq j \end{aligned} \quad (4.9)$$

If this transformation is used, all the Gerschgorin disks will be centered about the point  $1/c^2$ . In Reference 10, it has been shown that the eigenvalues of a symmetric, positive definite system are bounded above by  $\lambda_{\max} \leq n \bar{a}_{ii}$  where  $n$  is the maximum number of nonzero elements in any row.

One point which should be emphasized is that the scaling given by Equation (4.3) is generally unable to produce an "optimum" transformation. An optimum transformation is one which reduces  $C_N$  to its minimum value. The conditioning of some functions can actually be made worse by a transformation of the form given by Equation (4.3). However, a scaling of this type is often useful in achieving a meaningful improvement in the conditioning of the function (see Reference 10).

To illustrate the effects of the transformation on a two dimensional ellipsoid, the curves for  $f = 100$  in the original

and scaled coordinates are shown in Figure 3 for the equations

$$\begin{aligned} f &= \frac{1}{2} X^T \begin{bmatrix} 8 & 12 \\ 12 & 50 \end{bmatrix} X \\ &= \frac{1}{2} Y^T \begin{bmatrix} 1.0 & 0.6 \\ 0.6 & 1.0 \end{bmatrix} Y \\ &= 100 \end{aligned}$$

where

$$\begin{aligned} X &= DY \\ D &= \begin{bmatrix} \sqrt{2}/4 & 0.0 \\ 0.0 & \sqrt{2}/10 \end{bmatrix} \end{aligned}$$

Even though the principal axes of the ellipsoid  $f$  does not coincide with the axes of the coordinate system, the reduction of the eccentricity can be detected by visual inspection. The conditioning numbers for the original and transformed systems are 8.9 and 4.0 respectively.

So far in this discussion, the application of the scaling transformation has been made only to the well defined case of quadratic functions which describe ellipsoids in  $n$  space. What of the function  $\phi$  which is not quadratic? First, a portion of some ellipsoid can be approximately fitted onto the contour surface of the function  $\phi$  in the region of any given point. If it is assumed that the function and the ellipsoid are similar

in that region, then local scaling factors for  $\phi$  can be computed from Equation 4.9. These factors should at least improve the conditioning of the function in the local region of the point. As the region of the minimum is approached, it would be hoped that the scaling factors would converge to constant values. That is, in the vicinity of the minimum, the function should be closely approximated by the quadratic form of the Taylor series.

To investigate the effect of scaling on the function  $\phi$ , the three bar truss problem with only the stress constraints were studied in some detail. To accomplish this, the matrix of second partial derivatives was formed. Using this matrix, eigenvalues were found for several points. The results for two points are presented in Table 1. One point was an arbitrary, acceptable design and the other point was at the minimum of the function for a given  $v$  and  $W_0$ . As it can be seen, the scaling significantly reduced the value of  $C_N$  for both points.

Since the conditioning numbers for local regions of the function were reduced, it would appear that the minimization scheme might be able to work fairly effectively. This proved to be true. Using scaling, the minimization of the function with the Fletcher-Reeves method was able to proceed in a satisfactory manner.

One question which arose in dealing with the scaling factors was that of when to rescale or, at least, to check to see if rescaling was necessary. Since it was assumed that a scaling was

only valid for the local region of the point at which it was done, it was apparent that rescaling would probably have to be redone occasionally. To do it seldom would invite the possibility that, after three or four Fletcher-Reeves moves, the process would bog down until the next rescaling. This would be little better than the initial problem. Yet, to rescale too frequently would detract from the power of the Fletcher-Reeves scheme near the minimum since the direction vector  $\bar{s}$  should be replaced by the gradient after such a rescaling.

After running several tests with rescaling at various intervals, the occurrence of both problems was verified. A sequence of function values for rescaling every twenty steps for a thirty variable problem is shown in Figure 4. The first two scalings produce rapid decreases in the function value for the gradient moves. The third rescaling, done near the minimum, did not produce any great movement. In Figure 5, the effect of rescaling every seven iterations have been plotted for the same problem. In the latter stages of the process, it appears that the Fletcher-Reeves method is slowed by the rescaling. These results indicated that rescaling should be done often at the start of the minimization and less often towards the end of the process.

To accomplish these requirements, the following algorithm was developed. To begin the description of the algorithm, several variables should be defined. The variable  $C_b$  was the number of Fletcher-Reeves steps that the minimization was allowed to take

before the scaling factors were checked and changed if necessary. The counter  $C_a$  indicated the number of Fletcher-Reeves iterations since the last time the scaling factors were checked and  $C_d$  counted the number of times the scaling factors were checked and changed for some value of  $C_b$ . The flow diagram of the logic is shown in Figure 6. Briefly,  $C_b$  was set equal to one at the beginning of the minimization. Whenever  $C_a$  equaled  $C_b$ , the scaling factors were checked. If

$$\left| \frac{d_{ii,new} - d_{ii,old}}{d_{ii,old}} \right| < \Delta_c \quad i = 1, 2, \dots, n \quad (4.10)$$

was satisfied, then the old value of the scaling factors were retained,  $C_b$  was doubled,  $C_d$  was set to zero, and the next Fletcher-Reeves step was executed. If the condition given by Equation (4.10) was not satisfied, then the scaling factors were changed,  $C_d$  was incremented by one and the  $\bar{s}$  vector was replaced by the gradient. In either case,  $C_a$  was zeroed. Also, to reduce the chances of oscillation in the values of the scaling factors,  $C_b$  was increased by one if  $C_d$  managed to become equal to three. Such oscillations were observed to occur before the inclusion of this guard. The value of  $\Delta_c$  used in this study was 0.60.

## CHAPTER V

### RESULTS

A program incorporating the synthesis approach previously described was written and used to obtain numerical results for several test problems. A listing of the program can be found in Appendix B and the results for the test problems are summarized in this chapter.

The program was written in ALGOL 60 and was run on a UNIVAC 1107. The running times are presented in Table 2. Two times are reported for each test. The first time is the number of seconds which elapsed between the analysis of the initial design and the synthesis of the last analyzed design. The second is the time required for the program to realize that the goal weight was less than the optimum weight after the last analyzed design was found. It is estimated that these running times would be reduced by a factor of 4 to 7 if the programming were to be done in FORTRAN. This is a standard estimation based on comparisons of programs coded in both ALGOL and FORTRAN.

The times were presented to give some basis of the evaluation of the method's efficiency. However, caution should be used when drawing conclusions from these times.

The test problems included planar and three dimensional examples. For Case 1, a three bar truss, the optimum design was given in Reference 13. Also, results from References 1 and 8 of

Cases 2.A, 2.B, 3, and 4.A were available for comparison with the results obtained by this algorithm. The run times and final weights for both sets of results are given in Tables 15 and 16 respectively.

#### A. Planar Case

##### 1. Three Bar Truss

Consider a three bar truss as shown in Figure 1 subjected to the two independent load conditions  $P_1$  and  $P_2$ . If the constraints described in Chapter I are imposed and a high value of Young's modulus is selected, then the problem essentially becomes the same as an example presented in Reference 13 which has only two constraints

$$\sigma_{yt} - \sigma_i > 0$$

$$\sigma_i - \sigma_{yc} > 0$$

where

$$\sigma_{yt} = 20000 \text{ psi,}$$

$$\sigma_{yc} = -15000 \text{ psi}$$

For this problem the optimum weight is  $W_{opt} = 79.20$  which is given by the design  $A_1 = A_3 = 0.788 \text{ in}^2$  and  $A_2 = 0.41 \text{ in}^2$ . If the unit weight of the material is chosen to be  $1.0 \text{ lb/in}^3$ , then the optimum weight is  $79.2 \text{ lb}$ . Also, with the thickness preassigned as  $t_i = 1/4 \text{ in}$ , the diameter will be  $d_1 = d_3 = 3.15 \text{ in}$ . and



$d_2 = 1.64$  in. at the optimum design.

To assure that the displacement limits would not be active, the limits were given the large values of  $\pm 1.0$  in. Since displacements this large might violate the assumption of small displacements made in the formulation of the analysis, the displacements of the final design were checked. The largest displacement was less than 0.003 in. for an assumed value of  $3.0 \times 10^8$  psi for Young's modulus  $E$ . Hence, the validity of the assumption of small displacements was maintained.

The goal weight was selected to be 5% less than the weight of the last analyzed design. As can be seen from Table 3, the synthesis method was able to approximate the optimum within the expected amount of 5%. One interesting aspect of this problem is that since there are only two member design variables  $d_1$  and  $d_2$ , i.e.  $d_3 = d_1$ , the projection of the path of all the design variables can be made on a plane, i.e. the  $d_1, d_2$  plane. The path for Case 1 is shown in Figure 7. Designs which satisfy the analysis according to Equation (3.6) are denoted with circles while designs where minimum of the function  $\phi$  occurred and  $v$  was reduced are marked with triangles. Designs B and D were found by a proportional decrease in member diameters with an appropriate increase in the displacements. When such a modification could not be made so that the constraints were satisfied, the synthesis process was used to try to locate an analyzed design for the new goal weight. Design C is the result of the

synthesis algorithm. Design E is also the result of the process even though no analyzed design could be found. The synthesis was terminated after three minima were found for three decreasing values of  $v$  and a goal weight of 76.1 lb. That the analysis could not be satisfied became evident from an extrapolation for  $v = 0$  of the values of the residual  $R$  at these minima.

Figure 7 would seem to indicate that some of the intermediate designs between B and C did not satisfy the constraints. However, it must be remembered that the constraints shown in the figure are merely projections onto the  $d_1, d_2$  plane and must be satisfied only for analyzed designs. For example, any  $d_1$  and  $d_2$  which are positive would satisfy the stress and displacement constraints if all the displacements were set equal to zero. The design certainly would not be accurately analyzed but would satisfy the behavior constraints since all the member stresses would be zero.

## 2. Eleven Bar Truss

Consider the eleven bar truss shown in Figure 8 which is to be constructed of steel. The material properties of the steel are  $E = 3 \times 10^7$  psi and  $\rho = 0.28$  lb/in<sup>3</sup>. The three loading conditions to be applied to the structure are listed in Table 4. The stress limits include Euler buckling, crippling, tensile, and compressive yield stresses. The yield strength of the material was assumed to be  $\pm 50000$  psi. The design variables

used in this study were the member diameters. The thickness of all the tubular members were set to a value of  $1/10\pi$  or 0.0318 in.

Since the reactions are not symmetric, the design itself will not necessarily be symmetric despite the symmetry of the load conditions. Hence, linking of the design variables was not used in this study.

As in all other tests of this synthesis algorithm, the goal weight was selected to be 5% less than the weight of the last analyzed design. This was considered a reasonable reduction. After the final design has been produced using this reduction, the synthesis could be restarted at this last analyzed design for the finer tolerances of 1.0% or even 0.1% if the need presented itself.

The results for the displacement limits of  $\pm 0.15$  in. and  $\pm 0.25$  in. are presented in Tables 5 and 6 respectively. For the displacement limits of  $\pm 0.15$  in., Case 2.A, the design is limited by the displacement constraint in the downward direction at node 2 for the first load condition. The short time shown in Table 2 required to locate the final design is due to the fact that the final design was found from the initial, analyzed design by a simple proportional reduction of the design variables with the appropriate adjustments for the displacements. In this case the synthesis process was only used to determine that the weight of the design given by the scaling was at least within 5% of the

optimum weight.

The design with displacement limits of  $\pm 0.25$  in., Case 2.B, was essentially fully stressed. The final weight was 57.6 lb and the last goal weight was 54.7 lb. As indicated in Table 16, the optimum weight is near 55.5 lb. It was felt that the long time required to terminate the synthesis for this case was due to the fact that the goal weight was near the optimum.

From Table 16, the final weights given by this algorithm agree with the results given in Reference 1. However, for Case 2.A the values of the diameters are considerably different for the two designs. This could be expected since it has been pointed out by Reference 1 that for displacement limited problems, a number of designs of equal weight can be found so that all have the same value for a particular displacement. The designs for Case 2.B are fairly similar.

There would appear to be some reduction in the run times for the new algorithm. Without the termination time for the results of Reference 1, the over all efficiency of the methods can only be qualitatively stated. Beside being different in formulation, the two approaches were carried out with different minimization schemes.

## B. Three Dimensional Cases

### 1. Nine Bar Truss

Consider a nine bar space truss shown in Figure 9

subjected to the two load conditions given by Table 7. The material from which the structure is to be made is magnesium with  $E = 6.5 \times 10^6$  psi and  $\rho = 0.065$  lb/in<sup>3</sup>. The tensile and compressive yield of the material were 30000 psi and -17000 psi respectively. Both the thicknesses and diameters were allowed to vary and linking was imposed so that the groups  $d_1; t_1;$   
 $d_2 = d_3 = d_4 = d_5; t_2 = t_3 = t_4 = t_5; d_6 = d_7 = d_8 = d_9;$  and  
 $t_6 = t_7 = t_8 = t_9$  were formed to give a total of six member design variables.

Starting from the initial design weighing 10.92 lb, the final design for this structure, Case 3, was 5.97 lb for a displacement limit of  $\pm 10.0$  in. Again this limit was made artificially large so that the stress constraints would control the design. To assure that the assumption of small displacements was valid, the maximum displacement of the final design was checked and found to be less than 0.25 in. -- certainly small enough to satisfy the assumption. The sizes of the initial and final diameters and thicknesses for this problem are given in Table 8.

Again, comparable results are available for this test case. From Table 8, the final weights are quite close and the designs are almost the same. In this problem, the run time using the new algorithm was clearly good compared to that of the previous method.

## 2. Twenty Five Bar Truss

Consider the truss shown in Figure 10 which is to be constructed of aluminum with  $E = 10^7$  psi and  $\rho = 0.1$  lb/in<sup>3</sup>. The two load conditions are presented in Table 9. The material yield stress was taken to be  $\pm 40000$  psi. As usual the side constraints were placed on the member sizes but, notably, an upper limit of 100.0 was placed on the diameter to thickness ratio. The linking feature was employed so that the following groups of diameters were formed:  $d_1$ ;  $d_2 = d_3 = d_4 = d_5$ ;  $d_6 = d_7 = d_8 = d_9$ ;  $d_{10} = d_{11}$ ;  $d_{12} = d_{13}$ ;  $d_{14} = d_{15} = d_{16} = d_{17}$ ;  $d_{18} = d_{19} = d_{20} = d_{21}$ ; and  $d_{22} = d_{23} = d_{24} = d_{25}$ . The thicknesses were linked in a similar manner.

For Case 4.A the displacement limit was set at  $\pm 0.35$  in. The initial and final designs are presented in Table 10. The final design, which weighted 562.9 lb, was limited by the  $y$  displacement component of nodes 1 and 2 for both the load conditions. Also, the Euler buckling constraint for members 19 and 20 appeared to be critical for the second loading condition.

Comparison of design evolved by the new algorithm and the results given by Reference 8 shows a marked difference in the value of the diameters and thicknesses but a close agreement in the final weights. Again this is primarily due to the fact that the design is displacement limited. The run times are quite different with the new algorithm being the slower of the two. Part of the explanation is that the results for this problem presented in

Reference 8 were obtained using a variable metric minimization scheme. Such a scheme can be expected to complete the problem in about a third of the time required by the modified steepest descent minimization scheme used to obtain the results for Case 2.A, 2.B, and 3 in Reference 1.

The displacement limits were essentially removed by relaxing them to 10.0 in. and the synthesis for Case 4.B was initiated from a design weighing 581.3 lb, slightly heavier than the final design for Case 4.A. The final weight of 285.6 lb and other results for the case are given in Table 11. The active constraint was the upper limit of the diameter to thickness ratio for members 12 through 25. Also, the Euler buckling constraint was active for members 16 and 25 for the first load condition and members 19 and 20 for the second load condition. The magnitude of the greatest displacement is 0.805 in. for the  $y$  displacement at nodes 1 and 2 for the first load condition. From Table 11, the observation was made that members 10 through 13 tended to become small.

Starting from the final design of Case 4.B weighing 285.6 lb a final weight of 202.3 lb was obtained when the upper limit of the diameter to thickness ratio was raised to 400 for Case 4.C. The final design, given by Table 12, was stress limited in that at least one member of each group was bounded by the Euler buckling and crippling limits except for member 1. Member 1 was in tension for both loadings. The  $y$  displacement of nodes 1 and 2 gave the greatest displacement of 1.05 in. under the first load condition.

Again, there appears to be an attempt to eliminate members 10 through 13.

### 3. Thirty Bar Truss

Consider the dome shaped truss shown in Figure 11 to be made from steel with  $E = 30 \times 10^6$  psi and  $\rho = 0.28$  lb/in<sup>3</sup> to be Case 5. The yield strength of the material was taken to be  $\pm 40000$  psi. The two loading conditions imposed on the structure are given by Table 13. Linking of the thicknesses of members 1 through 6, 7 through 12, 13 through 18, and, finally, 19 through 30 was imposed. A similar design variable linking was made for the diameters.

A displacement limit of  $\pm 1.60$  in. was placed on the design. Starting from a design weighing 266600 lb a final design with a weight of 199700 lb was synthesized. The dimensions for the diameters and thicknesses for this problem are given in Table 14. The active constraints were the downward components of the displacements of nodes 4, 5, and 6 under the second load condition.



## CHAPTER VI

### DISCUSSION AND CONCLUSION

An algorithm was presented for finding the minimum weight design of a general, linear truss subject to a set of behavior and side constraints under an arbitrary number of imposed load conditions. This was accomplished by defining a goal weight and then minimizing the residuals subject to the original and weight constraints. Then, the Fiacco McCormick technique was used to transform this constrained minimization into one without constraints.

The ability of this algorithm to minimize the weight of the general truss was tested by considering several problems which have previously been solved using other algorithms. The results from these tests agreed with those from the other methods. From a comparison of run times, not including the termination times, it would appear that the new algorithm is usually faster in achieving a final design than similar methods. A comparison of the termination time, i.e. the time elapsed from the final design to the termination of the process, was not possible.

The formulation of  $\phi$  using the Fiacco McCormick method provided one clear advantage. Namely, the weight of the analyzed design produced by the synthesis was often considerably lower than the goal weight if the goal weight was much greater than the optimum. For the original algorithm which was developed within

framework of the integrated approach, the design weight was usually very close to the goal weight. In the formulation of that method, the problem was transformed into an unconstrained function by using an external penalty function which had a value only if a constraint was violated. However, since the Fiacco McCormick formulation usually took longer to find an analyzed design, only part of this advantage could be realized.

Some difficulties arose in the operation of the algorithm. These may generally be divided into two main areas. The first is concerned with the modification of an analyzed design to satisfy a new, lowered goal weight. The second is concerned with the long termination time. Hopefully, the following discussion can suggest some useful solutions to these difficulties.

Presently, an analyzed design is proportionally reduced so that all members experience the same percent reduction in their cross sectional area. Such a modification allows a very simple procedure to find the analysis of the design if all members are allowed to vary and no side constraints are active. Even though this modification has been found to be useful for the initial designs used for the test cases, it was evident that this method rarely provided any advantage in the latter stages of the synthesis. Another approach to the modification may be to extrapolate a new, modified design from the previous two analyzed design. This method would tend to account for the local shape of the constraint surfaces. Even though the advantage of

possibly obtaining another analyzed design from the current design is lost, the modified design from the proposed method may provide a better initial point in minimization of the function  $\phi$  and hence may have a tendency to shorten the run time.

The second major problem is the long termination time, (i.e. the time required for the program to realize that a goal weight is less than the optimum weight). Since the termination requires a succession of minimization for a sequence of  $v$ , this time depends greatly on the time to minimize  $\phi$ .

To reduce the time of the minimization, certain areas might be considered. First, an extrapolation of the minima have been developed for the Fiacco McCormick technique. Essentially, the extrapolation provides for an estimation of the minimum for a given  $v$  based on the minima for the previous two values of  $v$ . Even though the estimation seldom locates the minimum exactly, it usually will provide a significantly better starting point than the minimum for the previous  $v$ .

A second possibility in reducing the time for minimization is the selection of the criterion for the minimum. This difficulty is not unique to this method but the nature of this formulation does provide special considerations. Using the standard tests for the gradient and the change in the function value, the question of how stringently the criteria should be applied arises. It is felt that these criteria can be applied fairly loosely. Since a new minimization of  $\phi$  will be reinitiated after a

minimum is found and  $v$  is reduced, the extra effort required to find the minimum very accurately seems somewhat wasteful. The criteria does need to be harsh enough that the residual  $R$  will eventually be sufficiently reduced if the goal weight is greater than the optimum weight. Also, it would be desirable that a minimum will be accurate enough to serve as an adequate starting point for the minimization of  $\phi$  for the reduced value of  $v$ .

The formulation of the function  $\phi$  with a constrained goal weight suggests or allows the development of additional algorithms. One of these algorithms can be used to determine the feasibility of using particular configuration of a truss as a solution to a given structural problem. Generally, along with the loads and constraints, the feasibility study specifies a weight for which the truss may not be greater. Then the problem is one to find a design which satisfies the constraints and weighs less than the specified weight -- precisely the same problem which is solved over and over by the present algorithm. To incorporate this algorithm, the goal weight is replaced by the specified weight and a set of design variables are selected so as to satisfy the weight and side constraints. Then the displacements are set to zero, which also means that the stresses are zero, so that all the behavior constraints are satisfied. Finally, the process is initiated. If the truss is a feasible structural form for the problem, a design will be given.

Otherwise, the process will indicate that the particular configuration of the truss is an unfeasible solution.

A second additional algorithm which might be developed is one to find the optimum design by selecting a goal weight less than the optimum. Started in the same manner as the algorithm above, the process should indicate that the goal weight is less than the optimum weight. As a goal weight is shown to be unacceptable, a new, higher goal weight is selected until an analyzed design is found. Since Table 2 indicates that demonstrating a goal weight is unattainable takes considerable time, an effective form of an extrapolation would be required for the method to be efficient. An algorithm more efficient than the one presented by this paper might be developed if such an extrapolation could be found.

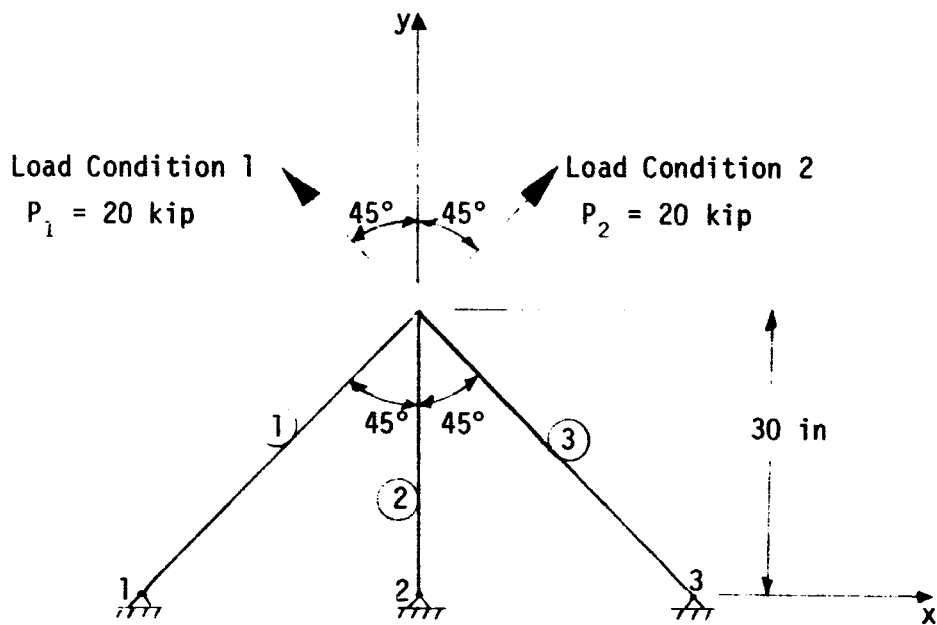


Figure 1 Three Bar Truss

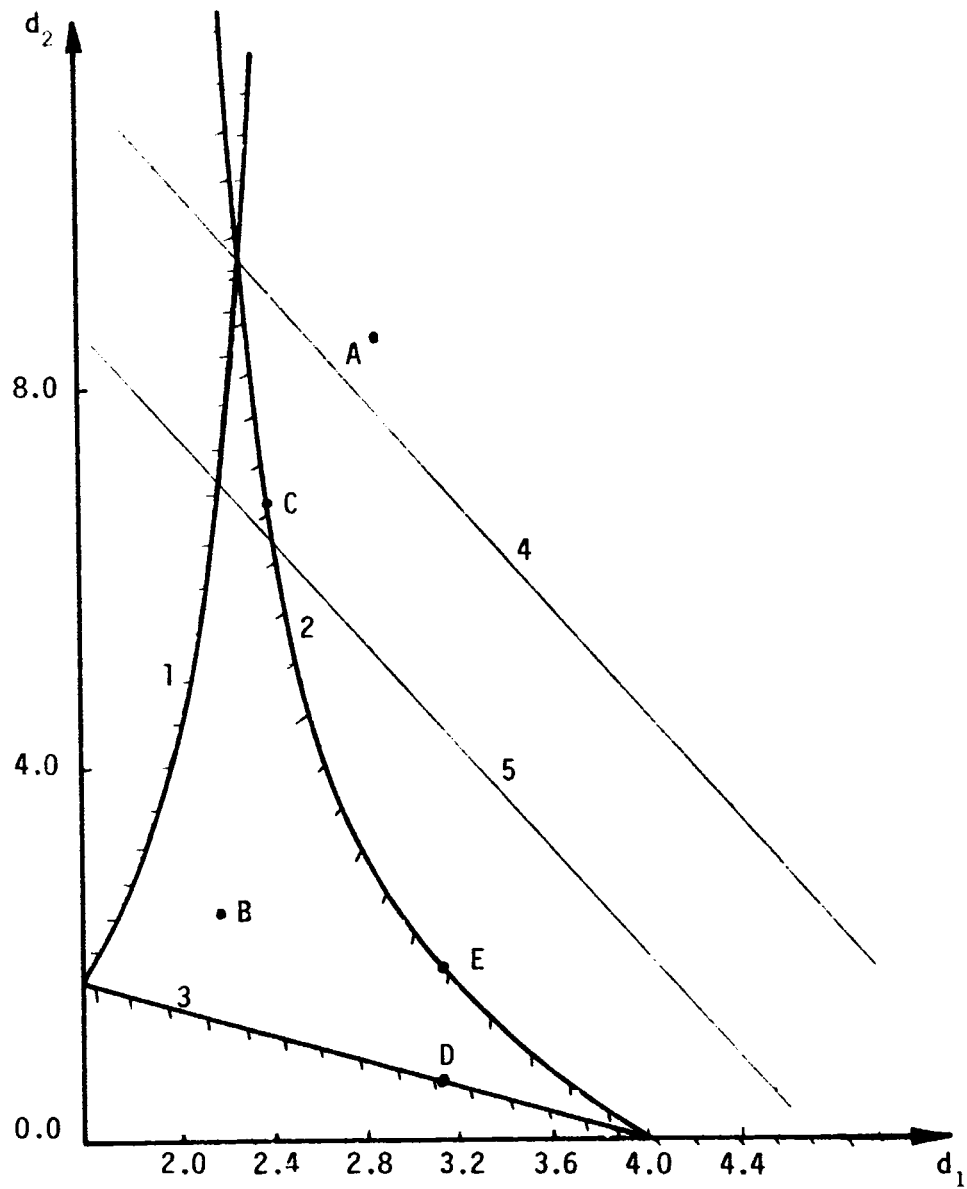


Figure 2 Design Space for Three Bar Truss

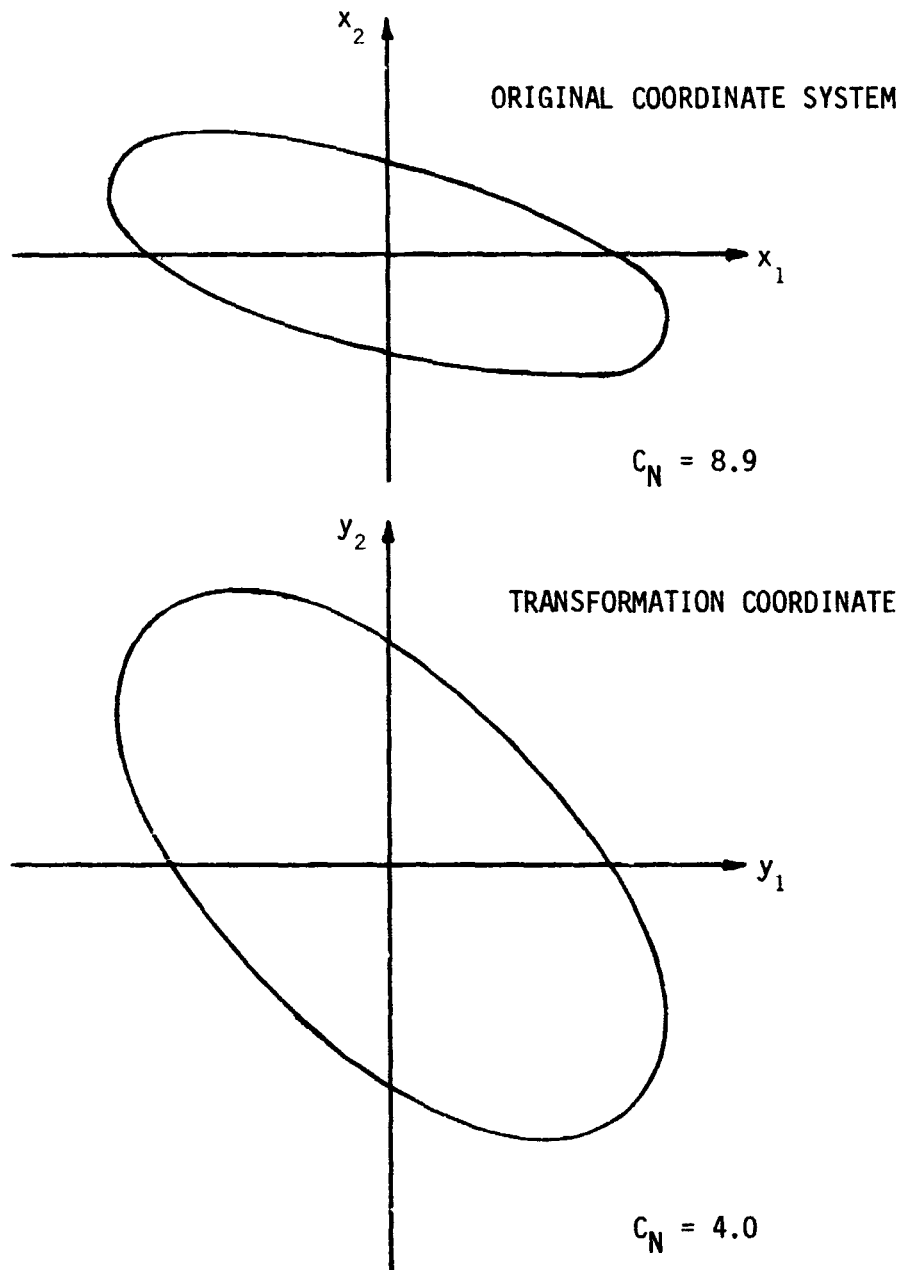


Figure 3 An Ellipsoid in its Original (x) and Transformed (y) Coordinate Systems



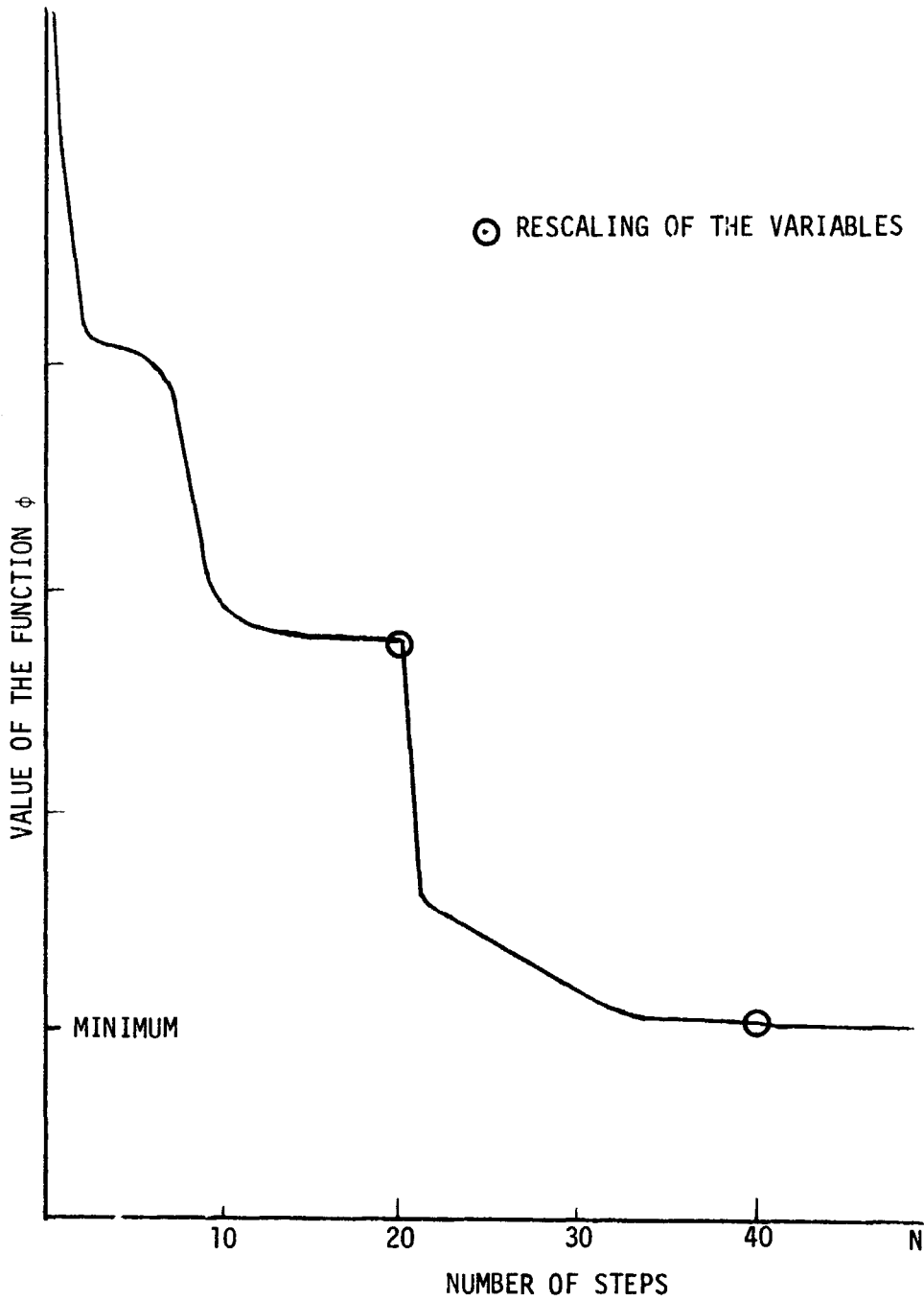


Figure 4 Value of  $\phi$  as a Function of the Number of Steps -- Rescaling Every 20 Steps

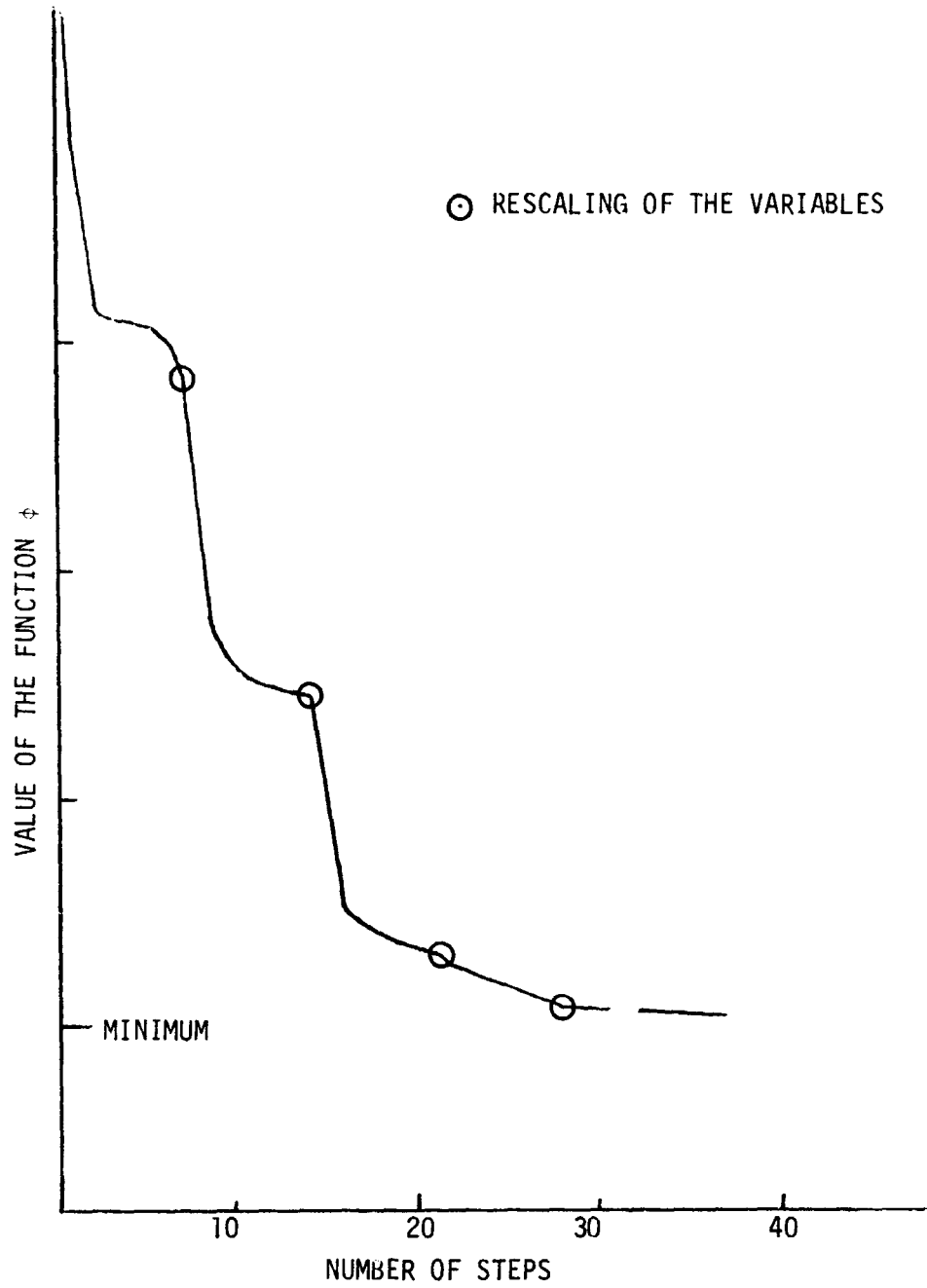


Figure 5 Value of  $\phi$  as a Function of the Number of Steps -- Rescaling Every 7 Steps

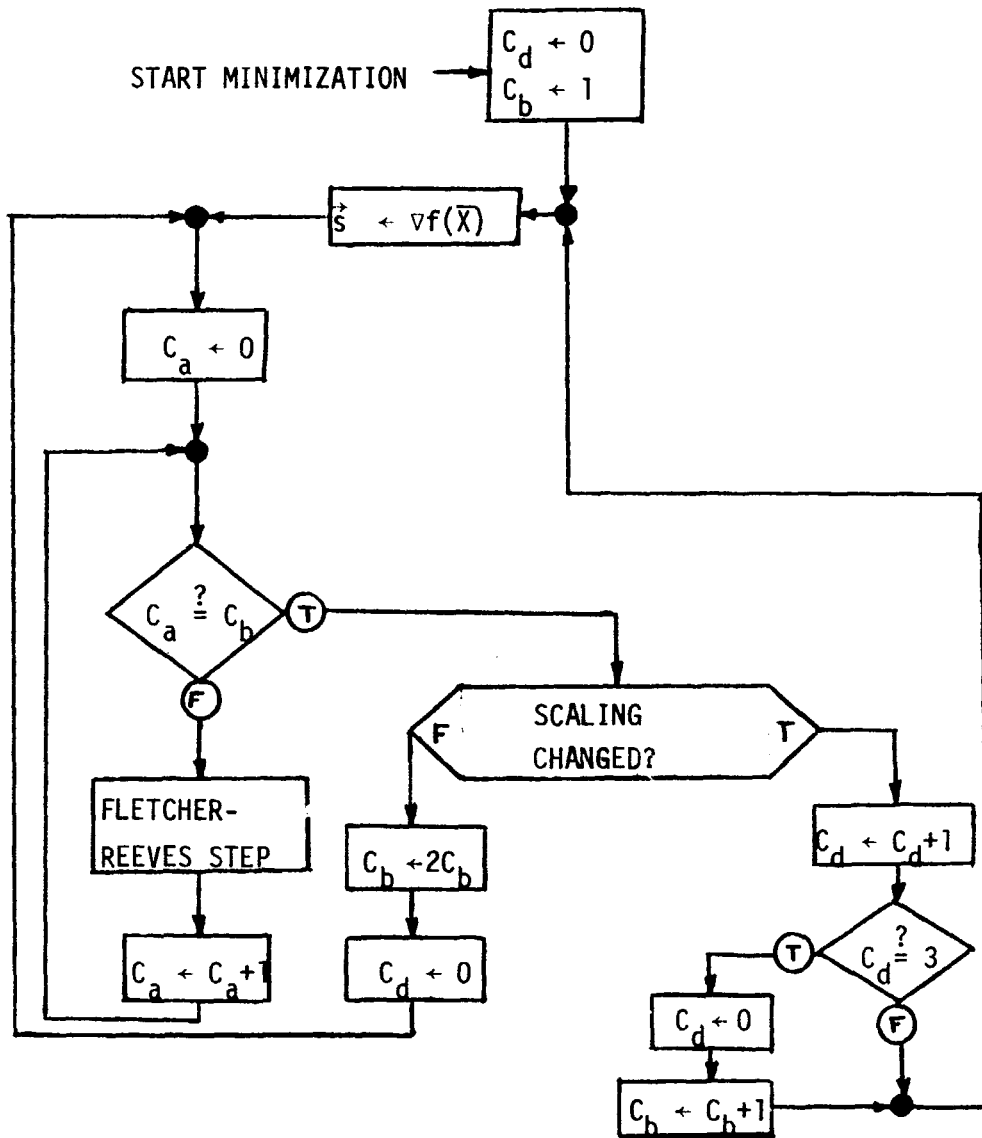


Figure 6 Flow Diagram for Frequency of Rescaling Logic

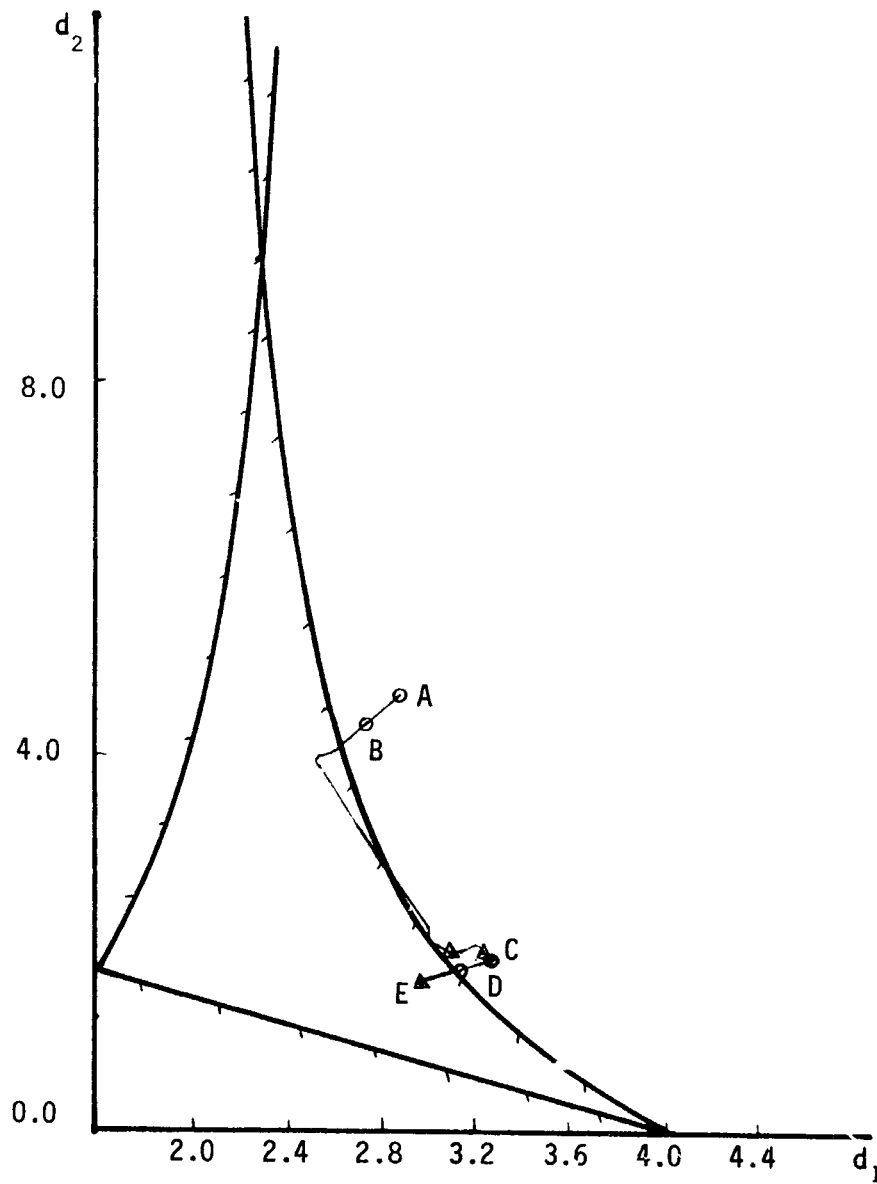


Figure 7 Synthesis Path for Three Bar Truss,  
Case 1

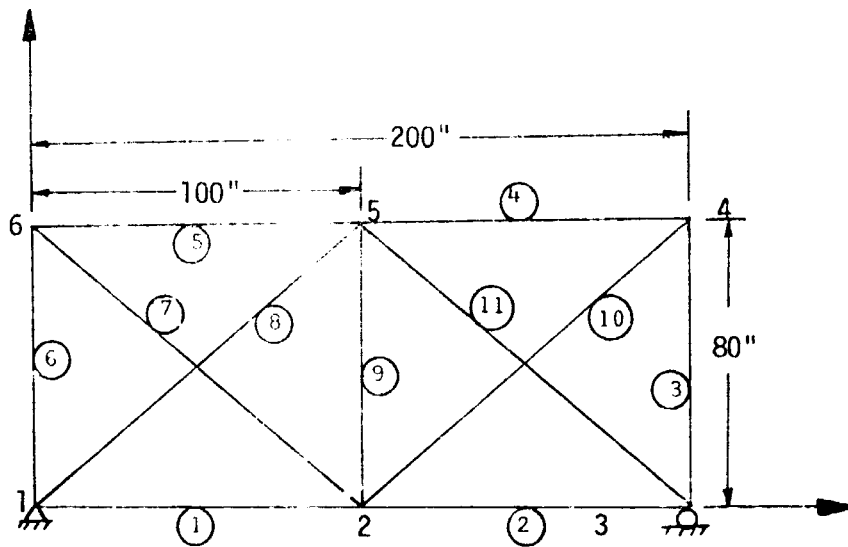


Figure 8 Eleven Bar Truss,  
Case 2

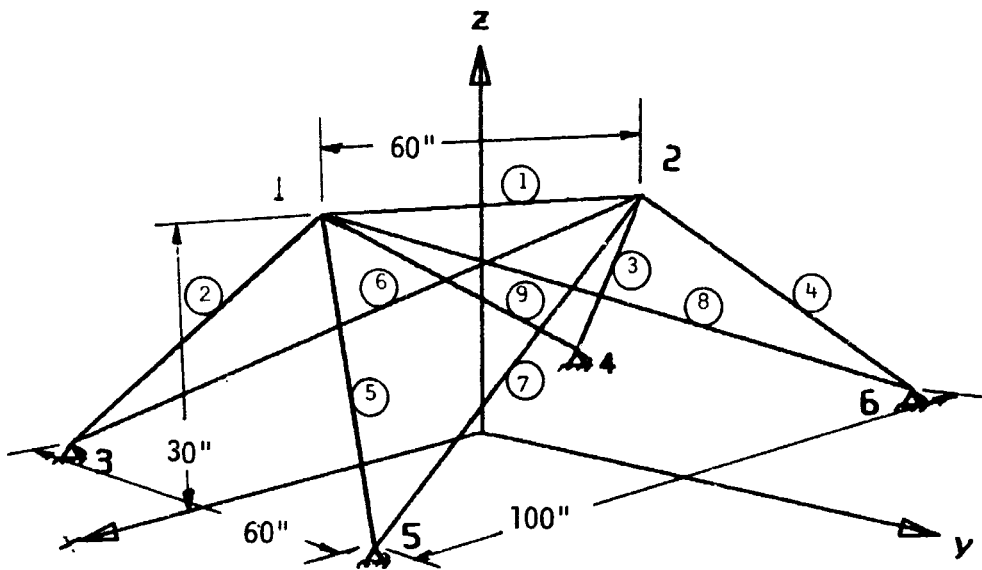


Figure 9 Nine Bar Truss,  
Case 3

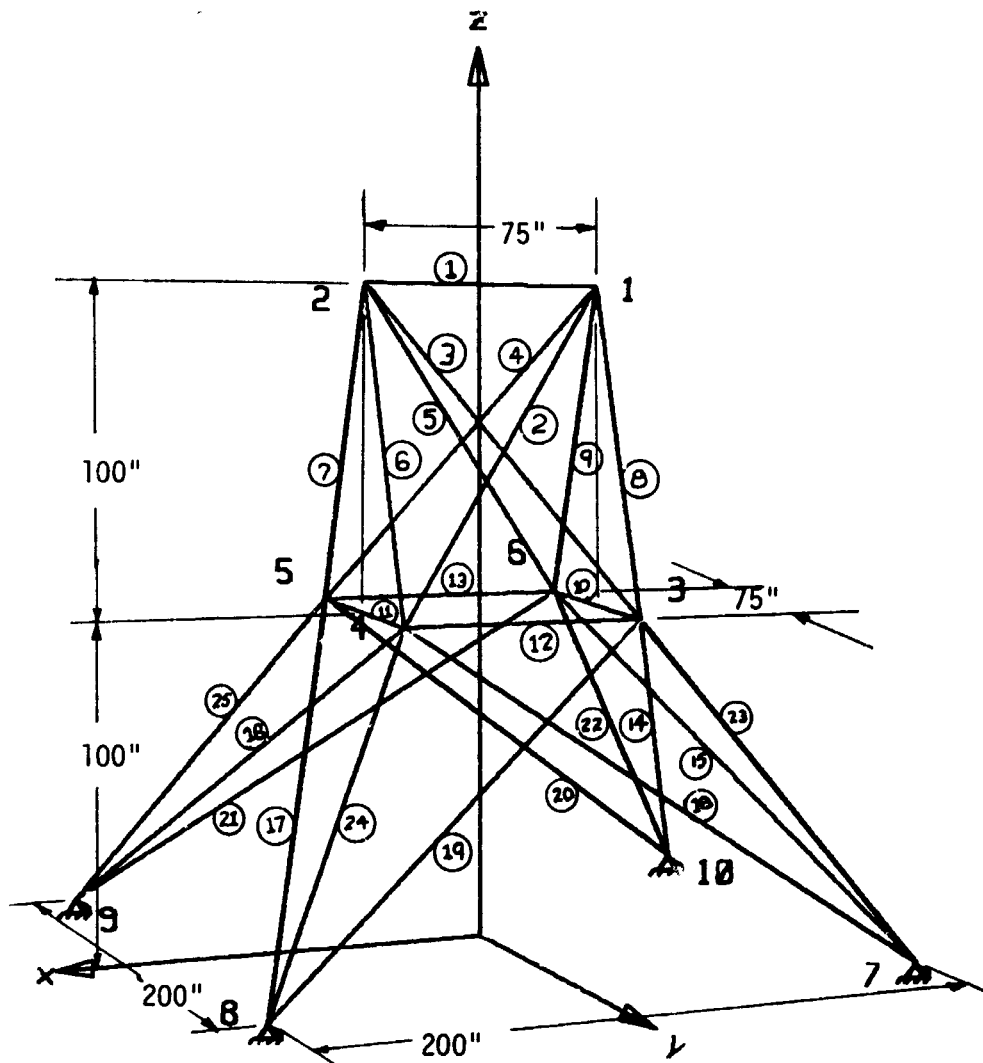


Figure 10 Twenty Five Bar Truss,  
Case 4

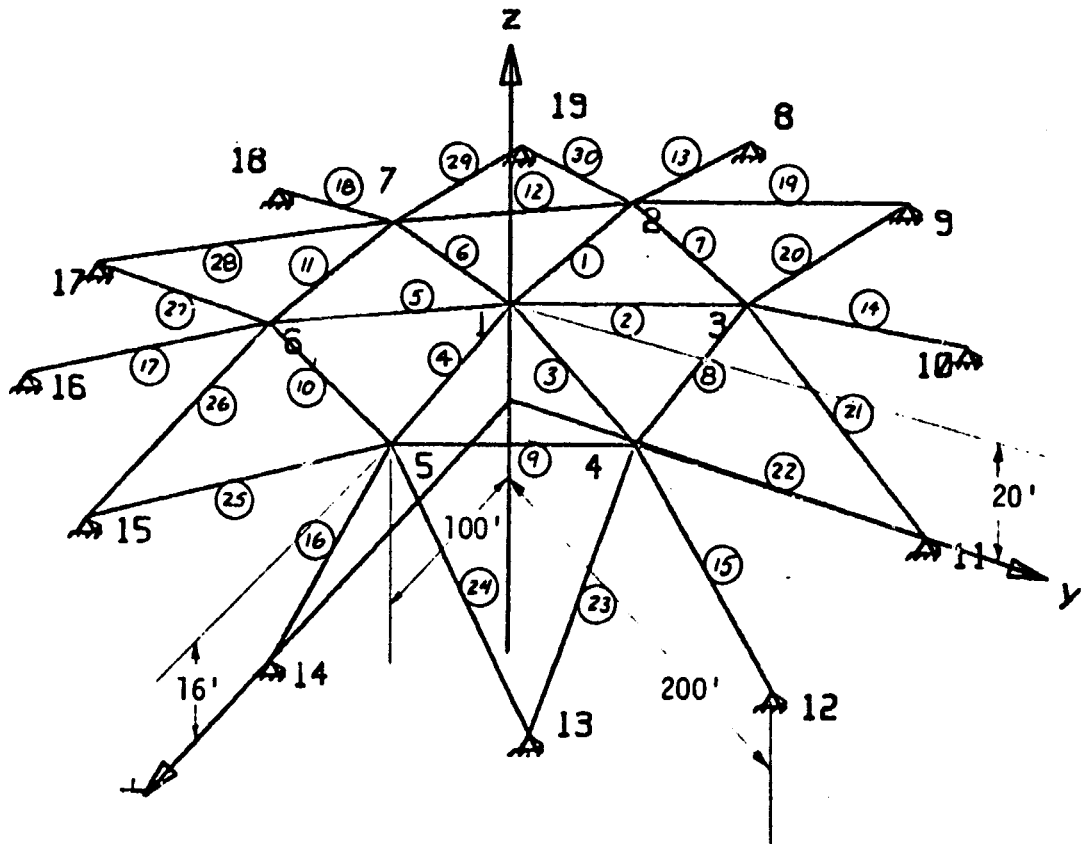


Figure 11 Thirty Bar Truss,  
Case 5



TABLE 1 EIGENVALUES FOR MATRIX OF SECOND PARTIAL  
DERIVATIVES FOR THREE BAR TRUSS

		Arbitrary Point		Minimum	
		Unscaled	Scaled	Unscaled	Scaled
Largest					
Eigenvalue	$\lambda_{\max}$	$9.95 \times 10^5$	2.33	$1.35 \times 10^6$	2.39
Smallest					
Eigenvalue	$\lambda_{\min}$	2.22	0.114	$1.54 \times 10^{-1}$	$6.46 \times 10^{-3}$
Conditioning					
Number	$C_N$	$4.46 \times 10^5$	20.4	$8.74 \times 10^6$	370

TABLE 2  
SUMMARY OF RUNNING TIMES

Case	Time to Find Final Design (Sec)	Termination Time (Sec)
1	38	30
2A	10	417
2B	505	1728
3	555	302
4A	1595	694
4B	2216	1230
4C	1973	1159
5	1040	2221

TABLE 3, CASE 1  
DISPLACEMENT LIMITS =  $\pm 1.0$  IN.

Member	Initial Diameters (in)	Final Diameters (in)	Optimum Design
1	2.88	3.148	3.15
2	4.64	1.729	1.64
3	2.88	3.148	3.15
Weight, lb	95.89	79.75	79.2

TABLE 4  
LOAD CONDITIONS FOR CASE 2

Load Condition 1  
(lbs)

$\vec{P}_{21}$	$\vec{P}_{41}$	$\vec{P}_{51}$	$\vec{P}_{61}$
-5000	-1000	-3000	1000
0	0	0	0

Load Condition 2  
(lbs)

$\vec{P}_{22}$	$\vec{P}_{42}$	$\vec{P}_{52}$	$\vec{P}_{62}$
0	0	0	0
0	3000	0	0

Load Condition 3  
(lbs)

$\vec{P}_{23}$	$\vec{P}_{43}$	$\vec{P}_{53}$	$\vec{P}_{63}$
0	0	0	0
0	0	0	-3000

TABLE 5, CASE 2.A  
DISPLACEMENT LIMITS =  $\pm 0.15$  IN.

Member	Initial Diameters (in)	Final Diameters (in)	Results From Ref. 1 (in)
1	4.00	3.095	3.87
2	3.10	2.399	2.44
3	3.49	2.700	2.50
4	3.42	2.646	2.50
5	3.27	2.530	2.22
6	3.14	2.430	2.00
7	2.96	2.290	2.92
8	3.61	2.793	3.28
9	2.22	1.702	1.26
10	3.55	2.747	3.00
11	3.20	2.476	2.65
Weight, lb	106.0	82.06	82.6

TABLE 6, CASE 2.B  
DISPLACEMENT LIMITS =  $\pm 0.25$  IN.

Member	Initial Diameters (in)	Final Diameters (in)	Results From Ref. 1 (in)
1	4.00	1.740	1.85
2	3.10	1.425	1.32
3	3.49	1.719	1.69
4	3.42	1.960	1.84
5	3.27	1.678	1.69
6	3.14	1.594	1.61
7	2.96	.889	1.08
8	3.61	2.950	2.66
9	2.22	1.125	1.01
10	3.55	1.521	1.49
11	3.20	2.552	2.50
Weight, lb	106.0	57.3	56.1

TABLE 7  
LOAD CONDITIONS FOR CASE 3

Load Condition 1 (lbs)		Load Condition 2 (lbs)	
$\vec{P}_{11}$	$\vec{P}_{21}$	$\vec{P}_{12}$	$\vec{P}_{22}$
0	0	0	4000
1700	-1700	0	0
0	0	0	-3000

TABLE 8, CASE 3  
DISPLACEMENT LIMITS =  $\pm 10.0$  IN

Member	Initial Design		Final Design		Results From Reference 1	
	Diameters (in)	Thicknesses (in)	Diameters (in)	Thicknesses (in)	Diameters (in)	Thicknesses (in)
1	2.80	0.010	2.058	0.0109	2.16	.012
2,3,4,5	1.10	0.079	1.929	0.0133	1.80	.018
6,7,8,9	3.33	0.030	3.535	0.0180	3.55	.016
Weight, lb	11.062		5.971		6.03	



TABLE 9  
LOAD CONDITIONS FOR CASE 4

Load Condition 1  
(lbs)

$\vec{p}_{11}$	$\vec{p}_{21}$	$\vec{p}_{31}$	$\vec{p}_{61}$
1000	0	500	500
10000	10000	0	0
-5000	-5000	0	0

Load Condition 2  
(lbs)

$\vec{p}_{12}$	$\vec{p}_{22}$	$\vec{p}_{32}$	$\vec{p}_{62}$
0	0	0	0
20000	-20000	0	0
-5000	-5000	0	0

TABLE 10, CASE 4.A  
DISPLACEMENT LIMITS =  $\pm 0.35$  IN

Member	Initial Design		Final Design		Results From Reference 8	
	Diameters (in)	Thicknesses (in)	Diameters (in)	Thicknesses (in)	Diameters (in)	Thicknesses (in)
1	3.96	0.20	0.710	0.0243	2.74	0.091
2,3,4,5	3.96	0.20	3.270	0.2113	3.18	0.24
6,7,8,9	3.96	0.20	3.449	0.2149	2.80	0.25
10,11	3.96	0.20	1.011	0.0182	1.72	0.017
12,13	3.96	0.20	1.145	0.0685	3.35	0.034
14,15,16,17	3.96	0.20	3.603	0.0697	3.21	0.091
18,19,20,21	3.96	0.20	3.881	0.1655	3.97	0.15
22,23,24,25	3.96	0.20	3.333	0.2535	2.91	0.28
Weight, lb	822.9		562.9		570.4	

TABLE 11, CASE 4.B

DISPLACEMENT LIMIT =  $\pm 10.0$  IN.

UPPER d/t RATIO LIMIT = 100.0

Member	Initial Design		Final Design	
	Diameters	Thicknesses	Diameters	Thicknesses
	(in)	(in)	(in)	(in)
1	.80	0.025	0.920	0.0393
2,3,4,5	3.30	0.220	4.189	0.0873
6,7,8,9	3.45	0.220	3.567	0.1346
10,11	1.05	0.020	0.691	0.0152
12,13	1.15	0.070	1.879	0.0216
14,15,16,17	3.65	0.070	4.242	0.0433
18,19,20,21	3.90	0.170	5.608	0.0574
22,23,24,25	3.35	0.260	5.020	0.0509
Weight, lb	581.3		285.6	

TABLE 12, CASE 4.C  
 DISPLACEMENT LIMIT =  $\pm 10.0$  IN  
 UPPER d/t RATIO LIMIT = 400.0

Member	Initial Design		Final Design	
	Diameters (in)	Thicknesses (in)	Diameters (in)	Thicknesses (in)
1	0.920	0.0393	0.852	0.0245
2,3,4,5	4.184	0.0873	5.164	0.0442
6,7,8,9	3.567	0.1346	4.055	0.0890
10,11	0.691	0.0152	0.699	0.0093
12,13	1.879	0.0216	1.700	0.0083
14,15,16,17	4.242	0.0433	5.275	0.0220
18,19,20,21	5.608	0.0574	6.690	0.0338
22,23,24,25	5.020	0.0509	5.472	0.0405
Weight, lb	285.6		202.3	

TABLE 13  
LOAD CONDITION FOR CASE 5

Load Condition 1 (ksi)						
$\vec{p}_{11}$	$\vec{p}_{21}$	$\vec{p}_{31}$	$\vec{p}_{41}$	$\vec{p}_{51}$	$\vec{p}_{61}$	$\vec{p}_{71}$
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0
-78.5	125.0	125.0	125.0	125.0	125.0	125.0

Load Condition 2 (ksi)						
$\vec{p}_{12}$	$\vec{p}_{22}$	$\vec{p}_{32}$	$\vec{p}_{42}$	$\vec{p}_{52}$	$\vec{p}_{62}$	$\vec{p}_{72}$
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0
59.0	187.5	187.5	187.5	0.0	0.0	0.0

TABLE 14, CASE 5  
 DISPLACEMENT LIMITS =  $\pm 1.60$  IN.

Member	Initial Design		Final Design	
	Diameters (in)	Thicknesses (in)	Diameters (in)	Thicknesses (in)
1 → 6	15.0	1.0	12.25	0.817
7 → 12	15.0	1.0	13.31	0.888
13 → 18	15.0	1.0	12.30	0.820
18 → 30	15.0	1.0	14.16	0.844
Weight.lb	266600		199700	

TABLE 15  
COMPARISON OF RUNNING TIMES

Case	For Results From Algorithm		From Previous Results (Ref. 1)
	To Achieve Final Design <sup>a</sup> (sec)	Termination Time <sup>a</sup> (sec)	To Achieve Final Design (sec)
2.A	10	417	221 <sup>b,c</sup>
2.B	505	1728	1532 <sup>b</sup>
3	555	302	3561 <sup>b</sup>
4.A	1595	694	363 <sup>d</sup>

- a. Using the Fletcher-Reeves minimization method with scaling.
- b. Using a modified steepest descent minimization scheme.
- c. Estimate termination time was 375 seconds.
- d. Using a variable metric minimization scheme.

TABLE 16  
COMPARISON OF RESULTS

Case	Final Weights Using New Algorithm (1b)	Final Weights Using Comparable Methods <sup>a</sup> (1b)	Lowest Estimate of Optimum Weight (1b)
1	79.75	-----	79.2 <sup>d</sup>
2.A	82.06	82.6 <sup>b</sup>	79.6 <sup>b</sup>
2.B	57.30	56.1 <sup>b</sup>	55.4 <sup>b</sup>
3	5.97	6.03 <sup>b</sup>	5.96 <sup>b</sup>
4.A	562.9	570.4 <sup>c</sup>	562.9

a. Running times for these results given in Table 15.

b. References 1 or 8.

c. Reference 8.

d. Reference 13.



## LIST OF REFERENCES

1. Fox, R. L., "An Integrated Approach to Engineering Synthesis and Analysis", Ph.D. Thesis, Engineering Division, Case Institute of Technology, Cleveland, Ohio, 1965.
2. Shanley, F. R., Weight-Strength Analysis of Aircraft Structures, (2nd Edition, Dover Publications, Inc., New York, 1960).
3. Gellatly, R. A., Gallagher, R. H., Luberacki, W. A., "Development of a Procedure for Automated Synthesis of Minimum Weight Structures", Air Force Contract AF 33(657)-11748, Preliminary Copy.
4. Schmit, L. A., and Mallett, R. H., "Structural Synthesis and Design Parameter Hierarchy," J. Struct. Div. Proc. Am. Soc. Civil Engrs. 89, No. ST4, 269-299, (August, 1963).
5. Micks, W. R., "Bibliography of Literature on Optimum Design of Structures and Related Topics", RM 2304, The Rand Corporation, Santa Monica, California, 1958.
6. Wasiutynski, Z. and Brandt, A., "The Present State of Knowledge in the Field of Optimum Design of Structures," Applied Mechanics Reviews, Vol. 16, No. 5, May 1963, pp. 341-350.
7. Gerard, G., "Optimum Structural Design Concepts for Aerospace Vehicles: Bibliography and Assessment", ARA Technical Report No. 272-2, March 1965.
8. Fox, R. L. and Schmit, L. A., "Advances in Integrated Approach to Structural Synthesis", J. of Spacecraft and Rockets, Vol. 3, No. 6, pp. 858-866, 1966.
9. Norris, C. H. and Wilbur, J. B., Elementary Structural Analysis, (2nd Edition, McGraw-Hill Book Company, New York, 1962).

10. Fox, R., and Stanton, E. L., "Developments in Structural Analysis by Direct Energy Minimization," Case Institute of Technology of Case Western Reserve University. To be published in the AIAA Journal, Summer, 1968.
11. Fletcher, R. and Reeves, C. M., "Function Minimization by Conjugate Gradients", British Computer J., Vol. 7, No. 2, pp. 149-154, 1964.
12. Fletcher, R. and Powell, M. J. D., "A Rapidly Convergent Descent Method for Minimization," British Computer J., Vol. 6, No. 2, pp. 163-168, 1963.
13. Schmit, L. A., "Comment on 'Completely Automatic Weight Minimization Method for High-Speed Digital Computers'", J. of Aircraft, Vol. 1, No. 6, pp. 375-377, 1964.

## APPENDIX A

## FORMULAS

This appendix contains the formulas for the gradient and the diagonal elements of the matrix of second partial derivatives for the function  $\phi$  presented in Chapter III.

A. Gradients

$$\begin{aligned} \frac{\partial \phi}{\partial d_i} = & 2 \sum_{k=1}^{KT} \sum_{\ell=1}^N \left[ \left( \sum_{j=1}^N K_{\ell j} u_{jk} - P_{\ell k} \right) \sum_{p=1}^N \left( \frac{\partial K_{\ell p}}{\partial d_i} u_{pk} \right) \right] \\ & + v \left\{ \frac{1}{(d_{\max} - d_i)^2} - \frac{1}{d_i^2} + \frac{1}{t_i(r_u - d_i/t_i)^2} \right. \\ & - \frac{1}{t_i(d_i/t_i - r_\ell)^2} + \sum_{k=1}^{KT} \left[ - \frac{2K_d \pi^2 E_i d_i}{8L_i^2 C_{z1} \left( \sigma_{ik} + \frac{\pi^2 E_i (d_i^2 + t_i^2)}{8L_i^2} \right)^2} \right. \\ & \left. \left. + \frac{K_d k_2 E_i t_i}{d_i^2 \left( \sigma_{ik} + \frac{k_2 E_i t_i}{d_i} \right)^2} \right] \right. \\ & \left. + \frac{\pi \rho W_m W_o L_i t_i}{(W_o - W)^2} \right\} \end{aligned}$$

$$\begin{aligned} \frac{\partial \phi}{\partial t_i} = & 2 \sum_{k=1}^{KT} \sum_{\ell=1}^N \left[ \left( \sum_{j=1}^N K_{\ell j} u_{jk} - P_{\ell k} \right) \sum_{p=1}^N \left( \frac{\partial K_{\ell p}}{\partial t_i} u_{pk} \right) \right] \\ & + v \left\{ \frac{1}{(t_{\max} - t_i)^2} + \frac{d_i}{t_i^2 \left( \frac{d_i}{t_i} - r_\ell \right)^2} \right\} \end{aligned}$$

$$\begin{aligned}
& - \frac{d_i}{t_i^2 \left( r_u - \frac{d_i}{t_i} \right)^2} \\
& + \frac{KT}{k_{zi}} \left[ \frac{2K_d \pi^2 E_i t_i}{8L_i^2 C_{z1} \left( \sigma_{ik} + \frac{\pi^2 E_i (d_i^2 + t_i^2)}{8L_i^2} \right)^2} \right. \\
& \left. - \frac{K_d k_2 E_i}{d_i \left( \sigma_{ik} + \frac{k_2 E_i t_i}{d_i} \right)^2} \right] \\
& + \frac{\pi \rho W_m W_o L_i d_i}{(W_o - W)^2} \left. \right\}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \phi}{\partial u_{jk}} &= 2 \sum_{i=1}^N K_{ji} \left( \sum_{\ell=1}^N K_{i\ell} u_{\ell k} - P_{ik} \right) \\
& + v \left\{ \sum_{j=1}^M \sum_{v=1}^6 n \left( \frac{j}{SR_{iv}} \right) \left( \frac{K_d E_i \gamma_{ij}}{L_i} \right) \left[ \right. \right. \\
& \quad \frac{1}{(\sigma_{yt} - \sigma_{ik})^2} - \frac{1}{(\sigma_{ik} - \sigma_{yc})^2} \\
& \quad - \frac{1}{C_{z1} \left( \sigma_{ik} + \frac{\pi^2 E_i (d_i^2 + t_i^2)}{8L_i^2} \right)^2} \\
& \quad \left. \left. - \frac{1}{\left( \sigma_{ik} + \frac{k_2 E_i t_i}{d_i} \right)^2} \right] \right\}
\end{aligned}$$

$$\left. + \frac{K_{uz}}{(u_{up} - u_{jk})^2} - \frac{K_{uz}}{(u_{jk} - u_{lo})^2} \right\}$$

### B. Second Partial Derivatives

$$\begin{aligned} \frac{\partial^2 \phi}{\partial d_i^2} = & 2 \sum_{k=1}^{KT} \sum_{\ell=1}^N \left( \sum_{j=1}^N u_{\ell j} \frac{\partial K_{\ell j}}{\partial d_i} \right)^2 \\ & + v \left\{ \frac{2}{(d_{\max} - d_i)^3} + \frac{2}{d_i^3} \right. \\ & + \frac{2}{t_i^2 (r_u - d_i/t_i)^3} + \frac{2}{t_i^2 (d_i/t_i - r_\ell)^3} \\ & + \sum_{k=1}^{KT} \left[ \frac{2K_d}{C_{z1} \left( \sigma_{ik} + \frac{\pi^2 E_i (d_i^2 + t_i^2)}{8L_i^2} \right)^3} \left( \frac{2\pi^2 E_i d_i}{8L_i^2} \right)^2 \right. \\ & - \frac{2K_d \pi^2 E_i}{8L_i^2 C_{z1} \left( \sigma_{ik} + \frac{\pi^2 E_i (d_i^2 + t_i^2)}{8L_i^2} \right)^2} \\ & + \frac{2K_d}{\left( \sigma_{ik} + \frac{k_2 E_i t_i}{d_i} \right)^3} \left( \frac{k_2 E_i t_i}{d_i^2} \right)^2 \\ & \left. - \frac{2K_d k_2 E_i t_i}{d_i^3 \left( \sigma_{ik} + \frac{k_2 E_i t_i}{d_i} \right)^2} \right] \end{aligned}$$

$$+ \left. \frac{2\pi^2 \rho^2 W_m W_o L_i^2 t_i^2}{(W_o - W)^3} \right\}$$

$$\begin{aligned} \frac{\partial^2 \phi}{\partial t_i^2} = & 2 \sum_{k=1}^{KT} \sum_{\ell=1}^N \left( \sum_{j=1}^N u_{\ell j} \frac{\partial K_{\ell j}}{\partial t_i} \right)^2 \\ & + v \left\{ \frac{2}{(t_{\max} - t_i)^3} + \frac{2}{(d_i/t_i - r_\ell)^3} \left( \frac{d_i}{t_i^2} \right)^2 \right. \\ & - \frac{2d_i}{t_i^3 (d_i/t_i - r_\ell)^2} + \frac{2}{(r_u - d_i/t_i)^3} \left( \frac{d_i}{t_i^2} \right)^2 \\ & + \frac{2d_i}{t_i^3 (r_u - d_i/t_i)^2} \\ & + \sum_{k=1}^{KT} \left[ \frac{2K_d}{C_{z1} \left( \sigma_{ik} + \frac{\pi^2 E_i (d_i^2 + t_i^2)}{8L_i^2} \right)^3} \left( \frac{2\pi^2 E_i t_i}{8L_i^2} \right)^2 \right. \\ & - \frac{2K_d \pi^2 E_i}{8L_i^2 C_{z1} \left( \sigma_{ik} + \frac{\pi^2 E_i (d_i^2 + t_i^2)}{8L_i^2} \right)^2} \\ & + \left. \frac{2K_d k_2^2 E_i^2}{d_i^2 \left( \sigma_{ik} + \frac{k_2 E_i t_i}{d_i} \right)^3} \right] \\ & + \left. \frac{2\pi^2 \rho^2 W_m W_o L_i^2 d_i^2}{(W_o - W)^3} \right\} \end{aligned}$$

$$\begin{aligned}
\frac{\partial^2 \phi}{\partial u_{jk}^2} = & 2K_{jj}^2 + v \left\{ \sum_{i=1}^M \sum_{v=1}^b \eta \left( \frac{j}{SR_{iv}} \right) \left( \frac{K_d E_i^2 \gamma_{iv}^2}{L_i^2} \right) \left[ \right. \right. \\
& \frac{2}{(\sigma_{\max} - \sigma_{ik})^3} + \frac{2}{(\sigma_{ik} - \sigma_{\min})^3} \\
& + \frac{2}{C_{z1} \left( \sigma_{ik} + \frac{\pi^2 E_i (d_i^2 + t_i^2)}{8L_i^2} \right)^3} \\
& \left. + \frac{2}{\left( \sigma_{ik} + \frac{k_2 E_i t_i}{d_i} \right)^3} \right] \\
& \left. + \frac{2K_{uz}}{(u_{up} - u_{jk})^3} + \frac{2K_{uz}}{(u_{jk} - u_{lo})^3} \right\}
\end{aligned}$$

C. The partial derivatives of the design and displacement variables were coordinated to the thickness, diameter, and displacements in the following manner:

#### Displacements

$$\frac{\partial \phi}{\partial X_i} = \frac{\partial \phi}{\partial u_{jk}} \qquad \frac{\partial^2 \phi}{\partial X_i^2} = \frac{\partial^2 \phi}{\partial u_{jk}^2}$$

$$i = j + nk$$

$$j = 1, 2, \dots, n$$

$$k = 1, 2, \dots, KT$$

Diameters

$$\frac{\partial \phi}{\partial \alpha_{i-nKT}} = \frac{\partial \phi}{\partial X_i} = \sum_{j=1}^M n \left( \frac{DD_j}{i-nKT} \right) \frac{\partial \phi}{\partial d_j}$$

$$\frac{\partial^2 \phi}{\partial \beta_{i-nKT}^2} = \frac{\partial^2 \phi}{\partial X_i^2} = \sum_{j=1}^M n \left( \frac{DD_j}{i-nKT} \right) \frac{\partial^2 \phi}{\partial d_j^2}$$

$$i = nKT + 1, \dots, nKT + N_{dv}$$

Thicknesses

$$\frac{\partial \phi}{\partial \beta_{i-nKT-N_{dv}}} = \frac{\partial \phi}{\partial X_i} = \sum_{j=1}^M n \left( \frac{TT_j}{i-nKT-N_{dv}} \right) \frac{\partial \phi}{\partial t_j}$$

$$\frac{\partial^2 \phi}{\partial \beta_{i-nKT-N_{dv}}^2} = \frac{\partial^2 \phi}{\partial X_i^2} = \sum_{j=1}^M n \left( \frac{TT_j}{i-nKT-N_{dv}} \right) \frac{\partial^2 \phi}{\partial t_j^2}$$

$$i = nKT + N_{dv} + 1, \dots, nKT + N_{dv} + N_{tv}$$



## APPENDIX B

The following are the listings of the ALGOL programs used to obtain the results presented in Chapter 5:

1. an ALGOL procedure, FP, to compute the value of the function  $\phi$ , its gradient, and the scaling factors;
2. the main program used to minimize the function  $\phi$  using the Fletcher-Reeves method.

```

1  A ALG FUNPHI
2  COMPILED BY CASE 1107 ALGOL 60 (FAT VERSION) DATED DECEMBER 14, 1967
3  THIS COMPIRATION WAS DONE ON 06 APR 68 AT 10:11:57
4  BU
5  BLOCK 1 LEVEL 1
6  BLOCK 2 LEVEL 2
7
8  1  PROCEDURE FP(ROW,X,DELX,PHEE,B2,B3, CHECK)S
9
10 2  INTEGER ROW,CHEK S
11 3  REAL PHEE S
12 4  REAL ARRAY X,DELX S
13 5  BOOLEAN B1,B2,B3 S
14
15 6  BEGIN
16 7  OWN INTEGER K1,N1,N2,NTV,NDV S
17 8  IF B1 THEN READ (K1,N1,N2,NTV,NDV) S
19 9  BEGIN
20 10 OWN REAL ARRAY KLOCAL(1:NM,1:J1,1:J2,1:J3),DRC(1:NM,1:J3),BUC,SMIN,SMAX,
21 11 KCONST,EA,D,IE,EALEN(LEN(LIM),P,U),KVS(0:NM,1:KI),
22 12 ALPHA(0:NM),BETATU(NTV),CONROU(1:NM,1:J3),
23 13 FSCALE(1:NM,K1+NDV+NTV) S
24 14 REAL ARRAY LITILE(UN),STRESS(1:NM,1:KT),F(1:0:1:2),Y(1:2,1:0),
25 15 LL(1:2,1:2),XDO(1:3),PARTU,PUR,PLUC,PLUCU,PLC,KLC(1:0),
26 16 MV(0:N) S
27 17 REAL ARRAY ISKAL,1:NM,K1+NDV+NTV S
28 18 OWN REAL NMU5,G11,G21,G31,G41,C1,C2,C3,C4,CZ11 S
29 19 OWN REAL RAL,RAU,DMAX,IMAX,1:J2S
30 20 OWN REAL CZ1,CZ2,D1R,MJ,NU,P1,P2,P3,PI,KMU,RNU,SAVPHI,SFC,
31 21 SFE,TNG,T5,T51,W,MW,MU,ULO,UUP,AAAPRR S
32 22 OWN INTEGER ARAT,SR1(1:16),PT(1:NM,1:2),OU(1:J3,NN),SRSM(1:0),
33 23 UD,IT(1:16) S
34 24 REAL AA,ALPH,BET,dB,CON1,CON2,CON3,CON4,CON5,CU10,CON7,
35 25 CON8,CON9,G1,G2,G3,G4,G5,G6,G15,G25,G35,G45,
36 26 G1C,G2C,G3C,G4C,MESS,P,SIG,TIT
37 27 OWN REAL WA,WB,WCE,MU1,MU2,INC S
38 28 INTEGER ALI,R,C,CHECK,1,10,11,J,K,KM,L,NO,PT1,PT2,ROM,
39 29 J1,J2,S
40 30 OWN INTEGER NK1,NK2,NK3,NK4,NK5,NK6,NK7,NK8,NK9,NK10,NK11,NK12,NK13,NK14,NK15,NK16,NK17,NK18,NK19,NK20,NK21,NK22,NK23,NK24,NK25,NK26,NK27,NK28,NK29,NK30,NK31,NK32,NK33,NK34,NK35,NK36,NK37,NK38,NK39,NK40,NK41,NK42,NK43,NK44,NK45,NK46,NK47,NK48,NK49,NK50,NK51,NK52,NK53,NK54,NK55,NK56,NK57,NK58,NK59,NK60,NK61,NK62,NK63,NK64,NK65,NK66,NK67,NK68,NK69,NK70,NK71,NK72,NK73,NK74,NK75,NK76,NK77,NK78,NK79,NK80,NK81,NK82,NK83,NK84,NK85,NK86,NK87,NK88,NK89,NK90,NK91,NK92,NK93,NK94,NK95,NK96,NK97,NK98,NK99,NK100,NK101,NK102,NK103,NK104,NK105,NK106,NK107,NK108,NK109,NK110,NK111,NK112,NK113,NK114,NK115,NK116,NK117,NK118,NK119,NK120,NK121,NK122,NK123,NK124,NK125,NK126,NK127,NK128,NK129,NK130,NK131,NK132,NK133,NK134,NK135,NK136,NK137,NK138,NK139,NK140,NK141,NK142,NK143,NK144,NK145,NK146,NK147,NK148,NK149,NK150,NK151,NK152,NK153,NK154,NK155,NK156,NK157,NK158,NK159,NK160,NK161,NK162,NK163,NK164,NK165,NK166,NK167,NK168,NK169,NK170,NK171,NK172,NK173,NK174,NK175,NK176,NK177,NK178,NK179,NK180,NK181,NK182,NK183,NK184,NK185,NK186,NK187,NK188,NK189,NK190,NK191,NK192,NK193,NK194,NK195,NK196,NK197,NK198,NK199,NK200,NK201,NK202,NK203,NK204,NK205,NK206,NK207,NK208,NK209,NK210,NK211,NK212,NK213,NK214,NK215,NK216,NK217,NK218,NK219,NK220,NK221,NK222,NK223,NK224,NK225,NK226,NK227,NK228,NK229,NK230,NK231,NK232,NK233,NK234,NK235,NK236,NK237,NK238,NK239,NK240,NK241,NK242,NK243,NK244,NK245,NK246,NK247,NK248,NK249,NK250,NK251,NK252,NK253,NK254,NK255,NK256,NK257,NK258,NK259,NK260,NK261,NK262,NK263,NK264,NK265,NK266,NK267,NK268,NK269,NK270,NK271,NK272,NK273,NK274,NK275,NK276,NK277,NK278,NK279,NK280,NK281,NK282,NK283,NK284,NK285,NK286,NK287,NK288,NK289,NK290,NK291,NK292,NK293,NK294,NK295,NK296,NK297,NK298,NK299,NK300,NK301,NK302,NK303,NK304,NK305,NK306,NK307,NK308,NK309,NK310,NK311,NK312,NK313,NK314,NK315,NK316,NK317,NK318,NK319,NK320,NK321,NK322,NK323,NK324,NK325,NK326,NK327,NK328,NK329,NK330,NK331,NK332,NK333,NK334,NK335,NK336,NK337,NK338,NK339,NK340,NK341,NK342,NK343,NK344,NK345,NK346,NK347,NK348,NK349,NK350,NK351,NK352,NK353,NK354,NK355,NK356,NK357,NK358,NK359,NK360,NK361,NK362,NK363,NK364,NK365,NK366,NK367,NK368,NK369,NK370,NK371,NK372,NK373,NK374,NK375,NK376,NK377,NK378,NK379,NK380,NK381,NK382,NK383,NK384,NK385,NK386,NK387,NK388,NK389,NK390,NK391,NK392,NK393,NK394,NK395,NK396,NK397,NK398,NK399,NK400,NK401,NK402,NK403,NK404,NK405,NK406,NK407,NK408,NK409,NK410,NK411,NK412,NK413,NK414,NK415,NK416,NK417,NK418,NK419,NK420,NK421,NK422,NK423,NK424,NK425,NK426,NK427,NK428,NK429,NK430,NK431,NK432,NK433,NK434,NK435,NK436,NK437,NK438,NK439,NK440,NK441,NK442,NK443,NK444,NK445,NK446,NK447,NK448,NK449,NK450,NK451,NK452,NK453,NK454,NK455,NK456,NK457,NK458,NK459,NK460,NK461,NK462,NK463,NK464,NK465,NK466,NK467,NK468,NK469,NK470,NK471,NK472,NK473,NK474,NK475,NK476,NK477,NK478,NK479,NK480,NK481,NK482,NK483,NK484,NK485,NK486,NK487,NK488,NK489,NK490,NK491,NK492,NK493,NK494,NK495,NK496,NK497,NK498,NK499,NK500,NK501,NK502,NK503,NK504,NK505,NK506,NK507,NK508,NK509,NK510,NK511,NK512,NK513,NK514,NK515,NK516,NK517,NK518,NK519,NK520,NK521,NK522,NK523,NK524,NK525,NK526,NK527,NK528,NK529,NK530,NK531,NK532,NK533,NK534,NK535,NK536,NK537,NK538,NK539,NK540,NK541,NK542,NK543,NK544,NK545,NK546,NK547,NK548,NK549,NK550,NK551,NK552,NK553,NK554,NK555,NK556,NK557,NK558,NK559,NK560,NK561,NK562,NK563,NK564,NK565,NK566,NK567,NK568,NK569,NK570,NK571,NK572,NK573,NK574,NK575,NK576,NK577,NK578,NK579,NK580,NK581,NK582,NK583,NK584,NK585,NK586,NK587,NK588,NK589,NK590,NK591,NK592,NK593,NK594,NK595,NK596,NK597,NK598,NK599,NK600,NK601,NK602,NK603,NK604,NK605,NK606,NK607,NK608,NK609,NK610,NK611,NK612,NK613,NK614,NK615,NK616,NK617,NK618,NK619,NK620,NK621,NK622,NK623,NK624,NK625,NK626,NK627,NK628,NK629,NK630,NK631,NK632,NK633,NK634,NK635,NK636,NK637,NK638,NK639,NK640,NK641,NK642,NK643,NK644,NK645,NK646,NK647,NK648,NK649,NK650,NK651,NK652,NK653,NK654,NK655,NK656,NK657,NK658,NK659,NK660,NK661,NK662,NK663,NK664,NK665,NK666,NK667,NK668,NK669,NK670,NK671,NK672,NK673,NK674,NK675,NK676,NK677,NK678,NK679,NK680,NK681,NK682,NK683,NK684,NK685,NK686,NK687,NK688,NK689,NK690,NK691,NK692,NK693,NK694,NK695,NK696,NK697,NK698,NK699,NK700,NK701,NK702,NK703,NK704,NK705,NK706,NK707,NK708,NK709,NK710,NK711,NK712,NK713,NK714,NK715,NK716,NK717,NK718,NK719,NK720,NK721,NK722,NK723,NK724,NK725,NK726,NK727,NK728,NK729,NK730,NK731,NK732,NK733,NK734,NK735,NK736,NK737,NK738,NK739,NK740,NK741,NK742,NK743,NK744,NK745,NK746,NK747,NK748,NK749,NK750,NK751,NK752,NK753,NK754,NK755,NK756,NK757,NK758,NK759,NK760,NK761,NK762,NK763,NK764,NK765,NK766,NK767,NK768,NK769,NK770,NK771,NK772,NK773,NK774,NK775,NK776,NK777,NK778,NK779,NK780,NK781,NK782,NK783,NK784,NK785,NK786,NK787,NK788,NK789,NK790,NK791,NK792,NK793,NK794,NK795,NK796,NK797,NK798,NK799,NK800,NK801,NK802,NK803,NK804,NK805,NK806,NK807,NK808,NK809,NK810,NK811,NK812,NK813,NK814,NK815,NK816,NK817,NK818,NK819,NK820,NK821,NK822,NK823,NK824,NK825,NK826,NK827,NK828,NK829,NK830,NK831,NK832,NK833,NK834,NK835,NK836,NK837,NK838,NK839,NK840,NK841,NK842,NK843,NK844,NK845,NK846,NK847,NK848,NK849,NK850,NK851,NK852,NK853,NK854,NK855,NK856,NK857,NK858,NK859,NK860,NK861,NK862,NK863,NK864,NK865,NK866,NK867,NK868,NK869,NK870,NK871,NK872,NK873,NK874,NK875,NK876,NK877,NK878,NK879,NK880,NK881,NK882,NK883,NK884,NK885,NK886,NK887,NK888,NK889,NK890,NK891,NK892,NK893,NK894,NK895,NK896,NK897,NK898,NK899,NK900,NK901,NK902,NK903,NK904,NK905,NK906,NK907,NK908,NK909,NK910,NK911,NK912,NK913,NK914,NK915,NK916,NK917,NK918,NK919,NK920,NK921,NK922,NK923,NK924,NK925,NK926,NK927,NK928,NK929,NK930,NK931,NK932,NK933,NK934,NK935,NK936,NK937,NK938,NK939,NK940,NK941,NK942,NK943,NK944,NK945,NK946,NK947,NK948,NK949,NK950,NK951,NK952,NK953,NK954,NK955,NK956,NK957,NK958,NK959,NK960,NK961,NK962,NK963,NK964,NK965,NK966,NK967,NK968,NK969,NK970,NK971,NK972,NK973,NK974,NK975,NK976,NK977,NK978,NK979,NK980,NK981,NK982,NK983,NK984,NK985,NK986,NK987,NK988,NK989,NK990,NK991,NK992,NK993,NK994,NK995,NK996,NK997,NK998,NK999,NK1000,NK1001,NK1002,NK1003,NK1004,NK1005,NK1006,NK1007,NK1008,NK1009,NK1010,NK1011,NK1012,NK1013,NK1014,NK1015,NK1016,NK1017,NK1018,NK1019,NK1020,NK1021,NK1022,NK1023,NK1024,NK1025,NK1026,NK1027,NK1028,NK1029,NK1030,NK1031,NK1032,NK1033,NK1034,NK1035,NK1036,NK1037,NK1038,NK1039,NK1040,NK1041,NK1042,NK1043,NK1044,NK1045,NK1046,NK1047,NK1048,NK1049,NK1050,NK1051,NK1052,NK1053,NK1054,NK1055,NK1056,NK1057,NK1058,NK1059,NK1060,NK1061,NK1062,NK1063,NK1064,NK1065,NK1066,NK1067,NK1068,NK1069,NK1070,NK1071,NK1072,NK1073,NK1074,NK1075,NK1076,NK1077,NK1078,NK1079,NK1080,NK1081,NK1082,NK1083,NK1084,NK1085,NK1086,NK1087,NK1088,NK1089,NK1090,NK1091,NK1092,NK1093,NK1094,NK1095,NK1096,NK1097,NK1098,NK1099,NK1100,NK1101,NK1102,NK1103,NK1104,NK1105,NK1106,NK1107,NK1108,NK1109,NK1110,NK1111,NK1112,NK1113,NK1114,NK1115,NK1116,NK1117,NK1118,NK1119,NK1120,NK1121,NK1122,NK1123,NK1124,NK1125,NK1126,NK1127,NK1128,NK1129,NK1130,NK1131,NK1132,NK1133,NK1134,NK1135,NK1136,NK1137,NK1138,NK1139,NK1140,NK1141,NK1142,NK1143,NK1144,NK1145,NK1146,NK1147,NK1148,NK1149,NK1150,NK1151,NK1152,NK1153,NK1154,NK1155,NK1156,NK1157,NK1158,NK1159,NK1160,NK1161,NK1162,NK1163,NK1164,NK1165,NK1166,NK1167,NK1168,NK1169,NK1170,NK1171,NK1172,NK1173,NK1174,NK1175,NK1176,NK1177,NK1178,NK1179,NK1180,NK1181,NK1182,NK1183,NK1184,NK1185,NK1186,NK1187,NK1188,NK1189,NK1190,NK1191,NK1192,NK1193,NK1194,NK1195,NK1196,NK1197,NK1198,NK1199,NK1200,NK1201,NK1202,NK1203,NK1204,NK1205,NK1206,NK1207,NK1208,NK1209,NK1210,NK1211,NK1212,NK1213,NK1214,NK1215,NK1216,NK1217,NK1218,NK1219,NK1220,NK1221,NK1222,NK1223,NK1224,NK1225,NK1226,NK1227,NK1228,NK1229,NK1230,NK1231,NK1232,NK1233,NK1234,NK1235,NK1236,NK1237,NK1238,NK1239,NK1240,NK1241,NK1242,NK1243,NK1244,NK1245,NK1246,NK1247,NK1248,NK1249,NK1250,NK1251,NK1252,NK1253,NK1254,NK1255,NK1256,NK1257,NK1258,NK1259,NK1260,NK1261,NK1262,NK1263,NK1264,NK1265,NK1266,NK1267,NK1268,NK1269,NK1270,NK1271,NK1272,NK1273,NK1274,NK1275,NK1276,NK1277,NK1278,NK1279,NK1280,NK1281,NK1282,NK1283,NK1284,NK1285,NK1286,NK1287,NK1288,NK1289,NK1290,NK1291,NK1292,NK1293,NK1294,NK1295,NK1296,NK1297,NK1298,NK1299,NK1300,NK1301,NK1302,NK1303,NK1304,NK1305,NK1306,NK1307,NK1308,NK1309,NK1310,NK1311,NK1312,NK1313,NK1314,NK1315,NK1316,NK1317,NK1318,NK1319,NK1320,NK1321,NK1322,NK1323,NK1324,NK1325,NK1326,NK1327,NK1328,NK1329,NK1330,NK1331,NK1332,NK1333,NK1334,NK1335,NK1336,NK1337,NK1338,NK1339,NK1340,NK1341,NK1342,NK1343,NK1344,NK1345,NK1346,NK1347,NK1348,NK1349,NK1350,NK1351,NK1352,NK1353,NK1354,NK1355,NK1356,NK1357,NK1358,NK1359,NK1360,NK1361,NK1362,NK1363,NK1364,NK1365,NK1366,NK1367,NK1368,NK1369,NK1370,NK1371,NK1372,NK1373,NK1374,NK1375,NK1376,NK1377,NK1378,NK1379,NK1380,NK1381,NK1382,NK1383,NK1384,NK1385,NK1386,NK1387,NK1388,NK1389,NK1390,NK1391,NK1392,NK1393,NK1394,NK1395,NK1396,NK1397,NK1398,NK1399,NK1400,NK1401,NK1402,NK1403,NK1404,NK1405,NK1406,NK1407,NK1408,NK1409,NK1410,NK1411,NK1412,NK1413,NK1414,NK1415,NK1416,NK1417,NK1418,NK1419,NK1420,NK1421,NK1422,NK1423,NK1424,NK1425,NK1426,NK1427,NK1428,NK1429,NK1430,NK1431,NK1432,NK1433,NK1434,NK1435,NK1436,NK1437,NK1438,NK1439,NK1440,NK1441,NK1442,NK1443,NK1444,NK1445,NK1446,NK1447,NK1448,NK1449,NK1450,NK1451,NK1452,NK1453,NK1454,NK1455,NK1456,NK1457,NK1458,NK1459,NK1460,NK1461,NK1462,NK1463,NK1464,NK1465,NK1466,NK1467,NK1468,NK1469,NK1470,NK1471,NK1472,NK1473,NK1474,NK1475,NK1476,NK1477,NK1478,NK1479,NK1480,NK1481,NK1482,NK1483,NK1484,NK1485,NK1486,NK1487,NK1488,NK1489,NK1490,NK1491,NK1492,NK1493,NK1494,NK1495,NK1496,NK1497,NK1498,NK1499,NK1500,NK1501,NK1502,NK1503,NK1504,NK1505,NK1506,NK1507,NK1508,NK1509,NK1510,NK1511,NK1512,NK1513,NK1514,NK1515,NK1516,NK1517,NK1518,NK1519,NK1520,NK1521,NK1522,NK1523,NK1524,NK1525,NK1526,NK1527,NK1528,NK1529,NK1530,NK1531,NK1532,NK1533,NK1534,NK1535,NK1536,NK1537,NK1538,NK1539,NK1540,NK1541,NK1542,NK1543,NK1544,NK1545,NK1546,NK1547,NK1548,NK1549,NK1550,NK1551,NK1552,NK1553,NK1554,NK1555,NK1556,NK1557,NK1558,NK1559,NK1560,NK1561,NK1562,NK1563,NK1564,NK1565,NK1566,NK1567,NK1568,NK1569,NK1570,NK1571,NK1572,NK1573,NK1574,NK1575,NK1576,NK1577,NK1578,NK1579,NK1580,NK1581,NK1582,NK1583,NK1584,NK1585,NK1586,NK1587,NK1588,NK1589,NK1590,NK1591,NK1592,NK1593,NK1594,NK1595,NK1596,NK1597,NK1598,NK1599,NK1600,NK1601,NK1602,NK1603,NK1604,NK1605,NK1606,NK1607,NK1608,NK1609,NK1610,NK1611,NK1612,NK1613,NK1614,NK1615,NK1616,NK1617,NK1618,NK1619,NK1620,NK1621,NK1622,NK1623,NK1624,NK1625,NK1626,NK1627,NK1628,NK1629,NK1630,NK1631,NK1632,NK1633,NK1634,NK1635,NK1636,NK1637,NK1638,NK1639,NK1640,NK1641,NK1642,NK1643,NK1644,NK1645,NK1646,NK1647,NK1648,NK1649,NK1650,NK1651,NK1652,NK1653,NK1654,NK1655,NK1656,NK1657,NK1658,NK1659,NK1660,NK1661,NK1662,NK1663,NK1664,NK1665,NK1666,NK1667,NK1668,NK1669,NK1670,NK1671,NK1672,NK1673,NK1674,NK1675,NK1676,NK1677,NK1678,NK1679,NK1680,NK1681,NK1682,NK1683,NK1684,NK1685,NK1686,NK1687,NK1688,NK1689,NK1690,NK1691,NK1692,NK1693,NK1694,NK1695,NK1696,NK1697,NK1698,NK1699,NK1700,NK1701,NK1702,NK1703,NK1704,NK1705,NK1706,NK1707,NK1708,NK1709,NK1710,NK1711,NK1712,NK1713,NK1714,NK1715,NK1716,NK1717,NK1718,NK1719,NK1720,NK1721,NK1722,NK1723,NK1724,NK1725,NK1726,NK1727,NK1728,NK1729,NK1730,NK1731,NK1732,NK1733,NK1734,NK1735,NK1736,NK1737,NK1738,NK1739,NK1740,NK1741,NK1742,NK1743,NK1744,NK1745,NK1746,NK1747,NK1748,NK1749,NK1750,NK1751,NK1752,NK1753,NK1754,NK1755,NK1756,NK1757,NK1758,NK1759,NK1760,NK1761,NK1762,NK1763,NK1764,NK1765,NK1766,NK1767,NK1768,NK1769,NK1770,NK1771,NK1772,NK1773,NK1774,NK1775,NK1776,NK1777,NK1778,NK1779,NK1780,NK1781,NK1782,NK1783,NK1784,NK1785,NK1786,NK1787,NK1788,NK1789,NK1790,NK1791,NK1792,NK1793,NK1794,NK1795,NK1796,NK1797,NK1798,NK1799,NK1800,NK1801,NK1802,NK1803,NK1804,NK1805,NK1806,NK1807,NK1808,NK1809,NK1810,NK1811,NK1812,NK1813,NK1814,NK1815,NK1816,NK1817,NK1818,NK1819,NK1820,NK1821,NK1822,NK1823,NK1824,NK1825,NK1826,NK1827,NK1828,NK1829,NK1830,NK1831,NK1832,NK1833,NK1834,NK1835,NK1836,NK1837,NK1838,NK1839,NK1840,NK1841,NK1842,NK1843,NK1844,NK1845,NK1846,NK1847,NK1848,NK1849,NK1850,NK1851,NK1852,NK1853,NK1854,NK1855,NK1856,NK1857,NK1858,NK1859,NK1860,NK1861,NK1862,NK1863,NK1864,NK1865,NK1866,NK1867,NK1868,NK1869,NK1870,NK1871,NK1872,NK1873,NK1874,NK1875,NK1876,NK1877,NK1878,NK1879,NK1880,NK1881,NK1882,NK1883,NK1884,NK1885,NK1886,NK1887,NK1888,NK1889,NK1890,NK1891,NK1892,NK1893,NK1894,NK1895,NK1896,NK1897,NK1898,NK1899,NK1900,NK1901,NK1902,NK1903,NK1904,NK1905,NK1906,NK1907,NK1908,NK1909,NK1910,NK1911,NK1912,NK1913,NK1914,NK1915,NK1916,NK1917,NK1918,NK1919,NK1920,NK1921,NK1922,NK1923,NK1924,NK1925,NK1926,NK1927,NK1928,NK1929,NK1930,NK1931,NK1932,NK1933,NK1934,NK1935,NK1936,NK1937,NK1938,NK1939,NK1940,NK1941,NK1942,NK1943,NK1944,NK1945,NK1946,NK1947,NK1948,NK1949,NK1950,NK1951,NK1952,NK1953,NK1954,NK1955,NK1956,NK1957,NK1958,NK1959,NK1960,NK1961,NK1962,NK1963,NK1964,NK1965,NK1966,NK1967,NK1968,NK1969,NK1970,NK1971,NK1972,NK1973,NK1974,NK1975,NK1976,NK1977,NK1978,NK1979,NK1980,NK1981,NK1982,NK1983,NK1984,NK1985,NK1986,NK1987,NK1988,NK1989,NK1990,NK1991,NK1992,NK1993,NK1994,NK1995,NK1996,NK1997,NK1998,NK1999,NK2000,NK2001,NK2002,NK2003,NK2004,NK2005,NK2006,NK2007,NK2008,NK2009,NK2010,NK2011,NK2012,NK2013,NK2014,NK2015,NK2016,NK2017,NK2018,NK2019,NK2020,NK2021,NK2022,NK2023,NK2024,NK2025,NK2026,NK2027,NK2028,NK2029,NK2030,NK2031,NK2032,NK2033,NK2034,NK2035,NK2036,NK2037,NK2038,NK2039,NK2040,NK2041,NK2042,NK2043,NK2044,NK2045,NK2046,NK2047,NK2048,NK2049,NK2050,NK2051,NK2052,NK2053,NK2054,NK2055,NK2056,NK2057,NK2058,NK2059,NK2060,NK2061,NK2062,NK2063,NK2064,NK2065,NK2066,NK2067,NK2068,NK2069,NK2070,NK2071,NK2072,NK2073,NK2074,NK2075,NK2076,NK2077,NK2078,NK2079,NK2080,NK2081,NK2082,NK2083,NK2084,NK2085,NK2086,NK2087,NK2088,NK2089,NK2090,NK2091,NK2092,N
```

```

40 FORMAT FM20( '15,AL,01$
41 FORMAT FM21( '15,AL,01$
42 FORMAT FM22( '15,AL,01$
43 FORMAT FM23( '15,AL,01$
44 FORMAT FM24( '15,AL,01$
45 FORMAT FM25( '15,AL,01$
46 FORMAT FM26( '15,AL,01$
47 FORMAT FM27( '15,AL,01$
48 FORMAT FM28( '15,AL,01$
49 FORMAT FM29( '15,AL,01$
50 FORMAT FM30( '15,AL,01$
51 FORMAT FM31( '15,AL,01$
52 FORMAT FM32( '15,AL,01$
53 FORMAT FM33( '15,AL,01$
54 FORMAT FM34( '15,AL,01$
55 FORMAT FM35( '15,AL,01$
56 FORMAT FM36( '15,AL,01$
57 FORMAT FM37( '15,AL,01$
58 FORMAT FM38( '15,AL,01$
59 FORMAT FM39( '15,AL,01$
60 FORMAT FM40( '15,AL,01$
61 FORMAT FM41( '15,AL,01$
62 FORMAT FM42( '15,AL,01$
63 FORMAT FM43( '15,AL,01$
64 FORMAT FM44( '15,AL,01$
65 FORMAT FM45( '15,AL,01$
66 FORMAT FM46( '15,AL,01$
67 FORMAT FM47( '15,AL,01$
68 FORMAT FM48( '15,AL,01$
69 FORMAT FM49( '15,AL,01$
70 FORMAT FM50( '15,AL,01$
71 FORMAT FM51( '15,AL,01$
72 FORMAT FM52( '15,AL,01$
73 FORMAT FM53( '15,AL,01$
74 FORMAT FM54( '15,AL,01$
75 FORMAT FM55( '15,AL,01$
76 FORMAT FM56( '15,AL,01$
77 FORMAT FM57( '15,AL,01$
78 FORMAT FM58( '15,AL,01$
79 FORMAT FM59( '15,AL,01$
80 FORMAT FM60( '15,AL,01$
81 FORMAT FM61( '15,AL,01$
82 FORMAT FM62( '15,AL,01$
83 FORMAT FM63( '15,AL,01$
84 FORMAT FM64( '15,AL,01$
85 FORMAT FM65( '15,AL,01$
86 FORMAT FM66( '15,AL,01$
87 FORMAT FM67( '15,AL,01$
88 FORMAT FM68( '15,AL,01$
89 FORMAT FM69( '15,AL,01$
90 FORMAT FM70( '15,AL,01$
91 FORMAT FM71( '15,AL,01$
92 FORMAT FM72( '15,AL,01$
93 FORMAT FM73( '15,AL,01$
94 FORMAT FM74( '15,AL,01$
95 FORMAT FM75( '15,AL,01$
96 FORMAT FM76( '15,AL,01$
97 FORMAT FM77( '15,AL,01$
98 FORMAT FM78( '15,AL,01$
99 FORMAT FM79( '15,AL,01$
100 FORMAT FM80( '15,AL,01$
101

```

```

102      META( T1(I)*T1(I)*SMAX(I)*SMIN(I),*
103      LIST L56(IU*10,DMAX,IMAX,RAU,RAL,SFE,SFC,VE,MU,KH,*)$
104      LIST L57( 1,FOR J = 1,2,3 DO CUO*(I - J))$
105      LIST L58(I,FOR J = 1,2,3 DO P(UUJ3*(I - J+UJ),K) )$
106      COMMENT THE VALUE OF THE RESIDUALS FOR NU EQUAL TO ZERO $
107      PROCEDURE EXTRAP (IFA,MU,RTFBS)
108          BOOLEAN TFA,TFBS
109          REAL MU,R$
110          BEGIN
111              REAL SP $
112              OWN REAL H1,H2,R3,N1,N2,N3,I1,I2,I3,S1,S2,A1,C,U,DEI $
113              IF NOT TFA THEN BEGIN
114                  R1 = R2 = R3 = N1 = N2 = N3 = S1 = S2 = A1 = C = U = U $
115                  GO TO DONE $      END$
116                  R3 = R2 $ R2 = R1 $ R1 = R $
117                  N3 = N2 $ N2 = N1 $ N1 = SORT( NU )$
118                  IF R2 EQL U,U THEN GO TO DONE $
119                  S2 = S1 $ S1 = (R2 - R1)/(N2 - N1)$
120                  C = U $      D = A $ A = R1 - S1*N1 $
121                  IF R3 NEQ U,U THEN BEGIN
122                      T2 = 1.0 $ T3 = SORT( NU )$      I1 = 1.0/T3 $
123                      DET = (T3*T3 - T2*T2)*T1 + (T1*T1 - T3*T3)*T2
124                      + (T2*T2 - T1*T1)*T3 $
125                      AA = ( T1*(T1*(T2*R3 - T3*R2) + (T3*T3*(T1*R2 - T2*R1)
126                          + T2*T2*(T3*R1 - T1*R3) ) / DEI ) $
127                      WRITE (AA)$
128                      END$
129                      SP = (-7.0*S2 + 29.0*S1)/24.0 $
130                      IF B NEQ 0.0 THEN
131                          IF B GTR C,ZC AND
132                          MIN (AA,R1 - SP*N1) GIR C,ZP
133                          THEN TFA = TRUE $
134                      WRITE (A,B,R,MU,C,ZC,SP,S1)$
135                      DONE $
136                      END$
137      REAL PROCEDURE SUS(G,J)$
138      REAL U$, I,INTEGER J $ BEGIN
139      IF G LSS 10.0**J THEN SUS = 0.0 ELSE SOS = G $
140      END$
141      COMMENT MODIFY THE DESIGN SO THAT IT WILL SATISFY A REDUCED GOAL WEIGHTS
142      PROCEDURE MODIF( U,ALPHA,RTA,MOWINC, M2STEP)$
143      REAL ARRAY U,ALPHA,BETA$      REAL MU,INC,STEP$
144      INTEGER M $
145      BEGIN
146          INTEGER I,J,K,I1,I0 $
147          INTEGER JU $
148          REAL A,B,I,S,T,S,W,R,S,C,S,AA,ALPHA,BEI $
149          BOOLEAN H1, B2 $
150          INTEGER ARRAY JU,JT(1:M)$

```

```

151 A = SONT(1.0 - INC)$ B = 1.0 - INC $
152 W = ESTEP = 0.0 $
153 FOR I = (1,1,N) DO BEGIN
154 IT = IT(1)$ ID = DO(1)$
155 R1 = 10 NEG 0 $ R2 = 1. NEG 0 $
156 J(1) = 10 $ J(11) = 11$
157 FOR J = (1,1,N-1) DO BEGIN
158 IF R2 AND IT EQL J(1) THEN R2 = FALSE $
159 IF R1 AND ID EQL J(1) THEN R1 = FALSE $
160 IF R1 AND R2 THEN BEGIN
161 ALPHA(ID) = A*ALPHA(ID)$
162 BETA(11) = A*BETA(11) END ELSE
163 IF R1 THEN BEGIN
164 US = B*ALPHA(ID)$ TS = RAL*BETA(11)*T(1)$
165 IF OS LEQ TS THEN ALPHA( ID) = (TS + ALPHA(ID)*D(1))/2.0
166 ELSE ALPHA(ID) = US $ END ELSE
167 IF R2 THEN BEGIN
168 IS = B*BETA(11)$ DS = ALPHA(ID)*J(1)/RAUS
169 IF TS LEQ OS THEN BETA(11) = (OS + BETA(11)*T(1))/2.0
170 ELSE BETA(11) = TS $
171 ALPH = ALPHA(1)*J(1)$
172 BET = BETA(11)*T(1)$
173 W = W + ALPH*BET*F(1)$
174 ESTEP = ESTEP + 1.0/(RAUS-ALPH/BET + 1.0/(ALPH/BET - RAL)
175 + 1.0/(DMAX - ALPH) + 1.0/(IMAX - BET) + 1.0/ALPH $
176 END$
177 FOR K = (1,1,KT) DO FOR J = (1,1,N) DO BEGIN
178 IF TS LEQ USP AND TS GIR ULO THEN U(J,K) = TS$
179 ESTEP = ESTEP + KUZ/(UUP - U(J,K) ) + KUZ/(U(J,K) - ULO)$
180 ENDS$
181 LAB**
182 WJ = 0. $
183 FOR I = (1,1,N) DO BEGIN
184 IT = IT(1)$ ID = ID(1) $
185 ALPH = ALPHA(ID)*J(1)$ BET = BETA(11)*T(1)$
186 BS = BUC(11)*(ALPH*ALPH + BET*BET)$
187 CS = U.40*E(1)*BET/ALPH $
188 LAC**
189 FOR K = (1,1,KT) DO BEGIN
190 AA = 0.0 $ FOR J = 1,2,3 DO
191 AA = AA + EQLZ(1,J)* (UISR(1,J,K)-U( SR(1,J+3),K))$
192 DS = MAX(SMIN,-BS,-CS)$
193 AA = 1.0001*AA $
194 IF AA LEQ DS OR AA GEQ SMAX(1) THEN BEGIN
195 TS = 0.96*MAX(DS/AA,SMAX(1)/AA)$
196 FOR J = (1,1,N) DO UISR(1,J,K) = TS*U( SR(1,J),K)$
197 J = 1 $
198 GO TO LAC $
199 GO TO LAC $
200 ESTEP = ESTEP + 50000*( 1.0/(SMAX(1) - AA) +
201 1.0/(AA-SMIN(1) + 1.0/( CS*( AA + BS)) +
202 1.0/(AA + CS ))$
203 ENDS$
204 IF JJ NEG 0 THEN BEGIN WRITE (J)$ GO TO LAB $ ENDS$

```

```

205 *RITE(ALPHA,BETA,UP,ESTP) $
206 NO = 1.005*MS
207 *RITE (W0) $
208 END $

209 COMMENT IF THIS IS THE FIRST TIME THROUGH THE PROCEDURE READ IN THE DATA
210 AND SET UP THE CONSTANTS WHICH WILL BE USED IN THE SYNTHESIS $
211 IF B1 THEN BEGIN
212 READ (KHU) $
213 READ (KUZ) $
214 READ (B5,B7) $
215 READ (B9) $
216 PI = 3.1415927 $
217 LL(1,1) = 1.0 $          LL(1,2) = -1.0 $
218 LL(2,1) = -1.0 $         LL(2,2) = 1.0 $
219 FOR I = (1,M) DO READ (LISTA) $
220 K = 0 $
221 FOR I = (1,M) DO BEGIN
222 READ (LISTH) $
223 J = 3*J $
224 IF V NEQ 0 THEN BEGIN A = K + 1 $ U(I,J-2) = K $ END $
225 IF A NEQ 0 THEN BEGIN A = K + 1 $ U(I,J-1) = K $ END $
226 IF L NEQ 0 THEN BEGIN A = K + 1 $ U(I,J) = K $ END $
227 END $
228 FOR K = (1,MKT) DO BEGIN
229 READ (J) $
230 FOR I = (1,IJ) DO BEGIN
231 READ (LPI, P2, P3) $
232 V = 3*LS $
233 PI(U(V-2), A) = P1 $
234 PI(U(V-1), A) = P2 $
235 PI(U(V), A) = P3 $
236 END $
237 FOR I = (U(1,MV)) DO READ (ALPHA(I)) $
238 FOR I = (U(1,MV)) DO READ (BETA(I)) $
239 PI1 = PI(1,1) $
240 PI2 = PI(1,2) $
241 G1 = COURU(PI1,1) - CUO(U(PI2,1)) $
242 G2 = COURU(PI1,2) - CUO(U(PI2,2)) $
243 G3 = COURU(PI1,3) - CUO(U(PI2,3)) $
244 LEN(I) = SORT( G1*G1 + G2*G2 + G3*G3 ) $
245 KCONST(I) = PI*E(I)/LEN(I) $
246 G1 = G1/LEN(I) $
247 G2 = G2/LEN(I) $
248 G3 = G3/LEN(I) $
249 DKC(I,1) = G1 $
250 DKC(I,2) = G2 $
251 DKC(I,3) = G3 $
252 CUN1 = G1*G1*CONST(I) $
253 CUN2 = G2*G2*CONST(I) $
254 CUN3 = G3*G3*CONST(I) $
255 CUN4 = G1*G2*CONST(I) $
256 CUN5 = G2*G3*CONST(I) $
257

```

```

254 C*NB = G3*G1*KCONST(I)$
255 KLOCAL(I,1,1) = CUN1 $
256 KLOCAL(I,2,2) = CUN2 $
257 KLOCAL(I,3,3) = CUN3 $
258 KLOCAL(I,1,2) = KLOCAL(I,2,1) = C0.4 $
259 KLOCAL(I,1,3) = KLOCAL(I,3,1) = CUN6 $
260 KLOCAL(I,2,3) = KLOCAL(I,3,2) = CUN5 $
261 ENDS $
262 HB = FALSE $
263 READ (SFE,SFC)$
264 READ (RAU,RAL)$
265 G1 = PI*PI $
266 W = 0 $
267 FOR I = (1,1,M) DO BEGIN
268   KCONST(I) = ALPHA( UD(I) ) * BETA( IT(I) ) * D(I) * T(I) $
269   PT1 = PT(I,1) $
270   PT2 = PT(I,2) $
271   FOR L = 1,2,3 DO BEGIN
272     SH(I,L) = UU(3*PT1 - 3 + L) $
273     SH(I,L,3) = UU(3*PT2 - 3 + L) $
274   END $
275   EBL(I) = E(I)/LEN(I) $
276   FOR J = 1,2,3 DO EBL(I,J) = EBL(I)*URC(I,J) $
277   WF(I) = PI*HRO*LEN(I)$
278   W = W + WF(I)*KCONST(I) $
279   DUC(I) = G1*EBL(I)/ID*5FE*LF,(I) $
280   ENDS $
281   WU = 1.0005*W $
282   WA = WB = W $
283   HB = FALSE $
284   NAT = NATI $
285   NATD = NATI + NOV $
286   NATDT = NATD + NTV $
287   FOR I = (1,1,N,T) DO X(I) = 0.0 $
288   FOR I = (1,1,NTV) DO X(NKTD + I) = HEIA(I) $
289   FOR I = (1,1,NOV) DO X(NKT + I) = ALPHA(I) $
290   C41 = 125 $
291   READ (UJW,ULU)$
292   READ (DMA,IMAX)$
293   READ (MXE)$
294   FOR I = (1,1,NKTD) DO ISCALE(I) = 1.0 $
295   THG = NU $
296   FOR I = (1,1,NK'TDT) DO X(I) = X(I)/ISCALE(I) $
297   READ (I)$
298   IF I GT 0 THEN READ (WA)$
299   IF I GT 0 THEN READ (UJTHG,WM,WO,X,TSSCALE) $
300   IF U9 THEN BEGIN
301     WRITE (FR11)$
302     FOR I = (1,1,M) DO WRITE (FR12,LS5)$
303     WRITE (FR13,LS6)$
304     WRITE (FR14)$
305     FOR I = (1,1,M) DO *RITE (FR15,LS7)$
306     *RITE (FR16)$
307   END $
308
325
326
327
328
329
330
331

```

```

312 FOR K = (1+1)*K(1) DO BEGIN
313   WRITE(FR1,KA)$
314   FOR I = (1+1)*NN DO WRITE (FR18,LSRIS)   ENDS
315   ENDS
316   CZ11 = 1.0/CZ1$
317   READ (INC)$
318   WRITE (EJ)$
319   ENDS
320   L14**
321   B10 = FALSE $
322   N01 = NU $
323   N02 = 50000*NU $
324   IF CHECK EQ 4 OR CHECK EQ 3 THEN WRITE (FOR I = (1+1)*NKID(1) DO
325     X(I)*TSKALE(I) )$
326   FOR K = (1+1)*K(1) DO BEGIN
327     FOR I = (1+1)*N DO
328       U(L1KA) = X(K) + 1.0*TSKALE(KN + I) $
329     ENDS
330   FOR I = (NKT + 1+1)*NKID(1) DO ALPHA(I-NKT) = X(I)*TSKALE(I)$
331   FOR I = (NKT + 1+1)*NKID(1) DO BETA(I-NKTID) = X(I)*TSKALE(I)$
332   B4 = B0 = FALSE $
333   L17**
334   IF B3 THEN FOR I = (1+1)*K(1) DO DELX(I) = 0.0 $
335   IF CHECK EQ 3 THEN BEGIN
336     B4 = TRUE $
337     FOR I = (1+1)*NKID(1) DO BEGIN
338       TSKALE(I) = TSKALE(I) $
339       TSKALE(I) = 0.0 $
340     ENDS
341   ENDS
342   ALT = 0.0 $
343   K = 0.0 $
344   K = 0.0 $
345   PHEE = 0.0 $
346   COMMENT COMPUTE THE HEIGHT OF THE DESIGN $
347   FOR I = (1+1)*N DO BEGIN
348     I1 = IT(I) $
349     I0 = UD(I) $
350     ALPHA = ALPHA(I1)*I1$
351     RET = BETA(I1)*I1$
352     KCONST(I) = ALPHA*RET $
353     W = W + KCONST(I)*I1$
354   ENDS
355   COMMENT CALCULATE THE RESIDUALS FOR THE DESIGN $
356   AA = BB = 0.0 $
357   FOR K = (1+1)*K(1) DO BEGIN
358     U0(K) = 0.0 $
359     FOR L = (1+1)*N DO KVL(L) = 0.0 $
360     FOR I = (1+1)*N DO BEGIN
361       FOR L = (1+1)*N DO LVC(L) = U0(K)*I1(K)$
362       FOR V = (1+1)*N DO BEGIN
363         TS = ACUMST(I1)*KLOCAL(1,V)*LVC(L) - LVC(L)*V +
364           KLOCAL(1,V)*LVC(L) - LVC(L)*V + KLOCAL(1,V)*V*
365

```







```

849      KN = N*(K-1) $
850      FOR I = (1,1,M) DO BEGIN
851          IT = IT(1) $
852          ID = DO(1) $
853          RET = RET(1)*T(I) $
854          ALPH = ALPHA(I)*DI(I) $
855          FOR L = (1,1,6) DO BEGIN
856              J = SR(I,L) $
857              PUR(L) = U(J,K) $
858              LOC(L) = RV(I,J,K) $
859              G1 = G2 = 0.0 $
860              FOR L2 = (1,1,3) DO BEGIN
861                  TS = TS1 = BB = 0.0 $
862                  FOR J = (1,1,3) DO BEGIN
863                      IF KLOCAL(I,L,J) NEQ 0.0 THEN BEGIN
864                          CON7 = KLOCAL(I,L,J)*PUR(J) $
865                          CON2 = KLOCAL(I,L,J)*PUR(J + 3) $
866                          AA = CON7 - CON2 $
867                          TS = TS + AA $
868                          G1 = G1 + CON7*CON7 + CON2*CON2 $
869                          H3 = BB + KLOCAL(I,L,J) * (LOC(J) - LOC(J+3)) $
870                      END $
871                      AA = CON7 - CON2 $
872                      TS1 = ALPHA*TS $
873                      TS = RET*TS $
874                      BB = KCONST(I)*BB $
875                      AA = LOC(L) $
876                      G3 = LOC(L+3) $
877                      PLOC(L) = TS*AA $
878                      PLOC(L+3) = -TS*G3 $
879                      PLC(L) = TS1*AA $
880                      PLC(L+3) = -TS1*G3 $
881                      KLC(L) = BB $
882                      KLC(L+3) = -BB $
883                      G2 = ALPHA*PLOC(L) $
884                      G1 = RET*PLOC(L) $
885                      FOR L2 = (1,1,6) DO BEGIN
886                          IF IT NEQ 0 THEN
887                              DELX( NKTD + IT) = JELX( NKTD + IT) + PLC(L)*RNU5 $
888                          IF ID NEQ 0 THEN
889                              DELX( NKTD + ID) = JELX( NKTD + ID) + PLOC(L)*RNU5 $
890                          J = SR(I,L) $
891                          IF J NEW 0 THEN
892                              DELX( KN + J ) = JELX( KN + J ) + KLC(L)*RNU5 $
893                              IF B4 THEN BEGIN
894                                  IF II NEW 0 THEN
895                                      TSCALE( NKTD + IT) = TSCALE( NKTD + IT) + RNU5*2/500000 $
896                                  IF ID NEW 0 THEN
897                                      TSCALE( NKTD + ID) = TSCALE( NKTD + ID) + RNU5*1/500000 $
898                              END $
899                          END $
900                      END $
901                  END $
902              END $
903          END $
904      END $
905  END $
906  END $
907  END $
908  END $
909  END $
910  END $
911  END $
912  END $
913  END $
914  END $
915  END $
916  END $
917  END $
918  END $
919  END $
920  END $
921  END $
922  END $
923  END $
924  END $
925  END $
926  END $
927  END $

```

```

524 IF B4 THEN BEGIN
525   FOR I = (0.1*PI) DO LITTLE(I) = 0.0 $
526   FOR I = (1.1*PI) DO BEGIN
527     FOR L = (1.1*PI) DO BEGIN
528       AA = KLOCAL(I,L)*KCONSL(I) $
529       U1 = SK(I,L) $
530       U2 = SR(I,L*PI) $
531       LITTLE(U1) = LITTLE(U1) + AA $
532       LITTLE(U2) = LITTLE(U2) + AA $
533     END $
534   END $
535   FOR K = (1.1*PI) DO LITTLE(I) = LITTLE(I)*LITTLE(I)+KUS $
536   FOR K = (1.1*PI) DO BEGIN
537     KN = N*(K-1) $
538     FOR I = (1.1*PI) DO
539       ISCALE(KN + I) = ISCALE(KN + I) + LITTLE(I) $
540     END $
541   END $
542   ALT = 0 $
543   IF NOT AS THEN BEGIN
544     MESS = 0.0 $
545     FOR I = (K1+1.1*K101) DO DELX(I) = 0.0 $
546     GO TO LIR $
547   END $
548   COMMENT CHECK THE DISPLACEMENT CONSTRAINTS $
549   FOR I = (1.1*PI) DO BEGIN
550     V = (K - 1)*NS
551     FOR J = (1.1*PI) DO BEGIN
552       G1 = U(J) - U(J,K) $
553       G2 = U(J,K) - U(L) $
554       G3 = G1*G1 $
555       G4 = G2*G2 $
556       G5 = G3+G4 $
557       G6 = G5*G5 $
558       IF G6 LEQ 0.0 OR G6C LEQ 0.0 THEN BEGIN
559         ALT = 1 $
560         GO TO RETREAT $
561       END $
562       PHEE = PHEE + KUS*U(I)*G1 + 1.0/G2 $
563       DELX(J+V) = DELX(J+V) + KUS*U(I)*G1 - 1.0/G2 $
564       IF B4 THEN
565         ISCALE(J+V) = ISCALE(J+V) + 2*KUS*U(I)*G1 + 1.0/G2 $
566       END $
567     END $
568   END $
569   FOR K = (1.1*PI) DO U(J,K) = 0.0 $
570   FOR I = (1.1*PI) DO BEGIN
571     IT = IT(I) $
572     IU = U(I) $
573     IS = IS(I) + U*U $
574     AA = BDR = 0.0 $
575     NU = NU(I) $
576     CUN1 = ALPHA(I)*U(I) $
577     CUN2 = BETA(I)*U(I) $
578     CUN3 = RUL(I)*U(I) + CUN1 - CUN1 + CUN2 + CUN2 $
579     CUN4 = U*U*E(I)*CUN2/CUN1 $
580   END $

```

114

```

083 C1 = 2*H(1)*C(1)
084 C2 = 2*H(1)*C(2)
085 C3 = CON4/CUN1
086 C4 = CON4/CUN2
087 FOR J = (1+J2) TO N5G14
088   U1 = F(1+J2)
089   PART(U1) = G1
090   PART(U1+3) = -U1
091   FOR K = (1+KAT) TO N5G14
092     AA = 0.0
093     FOR J = 1 TO J2
094       AA = AA + E2*(U1+J2) - (U1+J2+3)*C(1)
095     S10 = AA
096     U1 = S4AA(1) - S10
097     U2 = S10 - S4AA(1)
098     U3 = S10 + CON2
099     U4 = S10 + CON4
100     U1 = S(1+J2-1)
101     U2 = S(1+J2-1)
102     U3 = S(1+J2-1)
103     U4 = S(1+J2-1)
104     IF G1 LEQ U1 OR G2 LEQ U2 OR G3 LEQ U3 OR G4 LEQ U4
105       THEN NEG1
106       ALT = 2
107       NO = N(1)
108       GO TO 40
109     U11 = 1.0/G1
110     U12 = 1.0/G2
111     U13 = 1.0/G3
112     U14 = 1.0/G4
113     U15 = 1.0/G5
114     U16 = 1.0/G6
115     U17 = 1.0/G7
116     U18 = 1.0/G8
117     U19 = 1.0/G9
118     U20 = 1.0/G10
119     U21 = 1.0/G11
120     U22 = 1.0/G12
121     U23 = 1.0/G13
122     U24 = 1.0/G14
123     U25 = 1.0/G15
124     U26 = 1.0/G16
125     U27 = 1.0/G17
126     U28 = 1.0/G18
127     U29 = 1.0/G19
130     U30 = 1.0/G20
131     U31 = 1.0/G21
132     U32 = 1.0/G22
133     U33 = 1.0/G23
134     U34 = 1.0/G24
135     U35 = 1.0/G25
136     U36 = 1.0/G26
137     U37 = 1.0/G27
138     U38 = 1.0/G28
139     U39 = 1.0/G29
140     U40 = 1.0/G30
141     U41 = 1.0/G31
142     U42 = 1.0/G32
143     U43 = 1.0/G33
144     U44 = 1.0/G34
145     U45 = 1.0/G35
146     U46 = 1.0/G36
147     U47 = 1.0/G37
148     U48 = 1.0/G38
149     U49 = 1.0/G39
150     U50 = 1.0/G40
151     U51 = 1.0/G41
152     U52 = 1.0/G42
153     U53 = 1.0/G43
154     U54 = 1.0/G44
155     U55 = 1.0/G45
156     U56 = 1.0/G46
157     U57 = 1.0/G47
158     U58 = 1.0/G48
159     U59 = 1.0/G49
160     U60 = 1.0/G50
161     U61 = 1.0/G51
162     U62 = 1.0/G52
163     U63 = 1.0/G53
164     U64 = 1.0/G54
165     U65 = 1.0/G55
166     U66 = 1.0/G56
167     U67 = 1.0/G57
168     U68 = 1.0/G58
169     U69 = 1.0/G59
170     U70 = 1.0/G60
171     U71 = 1.0/G61
172     U72 = 1.0/G62
173     U73 = 1.0/G63
174     U74 = 1.0/G64
175     U75 = 1.0/G65
176     U76 = 1.0/G66
177     U77 = 1.0/G67
178     U78 = 1.0/G68
179     U79 = 1.0/G69
180     U80 = 1.0/G70
181     U81 = 1.0/G71
182     U82 = 1.0/G72
183     U83 = 1.0/G73
184     U84 = 1.0/G74
185     U85 = 1.0/G75
186     U86 = 1.0/G76
187     U87 = 1.0/G77
188     U88 = 1.0/G78
189     U89 = 1.0/G79
190     U90 = 1.0/G80
191     U91 = 1.0/G81
192     U92 = 1.0/G82
193     U93 = 1.0/G83
194     U94 = 1.0/G84
195     U95 = 1.0/G85
196     U96 = 1.0/G86
197     U97 = 1.0/G87
198     U98 = 1.0/G88
199     U99 = 1.0/G89
200     U100 = 1.0/G90
201     U101 = 1.0/G91
202     U102 = 1.0/G92
203     U103 = 1.0/G93
204     U104 = 1.0/G94
205     U105 = 1.0/G95
206     U106 = 1.0/G96
207     U107 = 1.0/G97
208     U108 = 1.0/G98
209     U109 = 1.0/G99
210     U110 = 1.0/G100
211     U111 = 1.0/G101
212     U112 = 1.0/G102
213     U113 = 1.0/G103
214     U114 = 1.0/G104
215     U115 = 1.0/G105
216     U116 = 1.0/G106
217     U117 = 1.0/G107
218     U118 = 1.0/G108
219     U119 = 1.0/G109
220     U120 = 1.0/G110
221     U121 = 1.0/G111
222     U122 = 1.0/G112
223     U123 = 1.0/G113
224     U124 = 1.0/G114
225     U125 = 1.0/G115
226     U126 = 1.0/G116
227     U127 = 1.0/G117
228     U128 = 1.0/G118
229     U129 = 1.0/G119
230     U130 = 1.0/G120
231     U131 = 1.0/G121
232     U132 = 1.0/G122
233     U133 = 1.0/G123
234     U134 = 1.0/G124
235     U135 = 1.0/G125
236     U136 = 1.0/G126
237     U137 = 1.0/G127
238     U138 = 1.0/G128
239     U139 = 1.0/G129
240     U140 = 1.0/G130
241     U141 = 1.0/G131
242     U142 = 1.0/G132
243     U143 = 1.0/G133
244     U144 = 1.0/G134
245     U145 = 1.0/G135
246     U146 = 1.0/G136
247     U147 = 1.0/G137
248     U148 = 1.0/G138
249     U149 = 1.0/G139
250     U150 = 1.0/G140
251     U151 = 1.0/G141
252     U152 = 1.0/G142
253     U153 = 1.0/G143
254     U154 = 1.0/G144
255     U155 = 1.0/G145
256     U156 = 1.0/G146
257     U157 = 1.0/G147
258     U158 = 1.0/G148
259     U159 = 1.0/G149
260     U160 = 1.0/G150
261     U161 = 1.0/G151
262     U162 = 1.0/G152
263     U163 = 1.0/G153
264     U164 = 1.0/G154
265     U165 = 1.0/G155
266     U166 = 1.0/G156
267     U167 = 1.0/G157
268     U168 = 1.0/G158
269     U169 = 1.0/G159
270     U170 = 1.0/G160
271     U171 = 1.0/G161
272     U172 = 1.0/G162
273     U173 = 1.0/G163
274     U174 = 1.0/G164
275     U175 = 1.0/G165
276     U176 = 1.0/G166
277     U177 = 1.0/G167
278     U178 = 1.0/G168
279     U179 = 1.0/G169
280     U180 = 1.0/G170
281     U181 = 1.0/G171
282     U182 = 1.0/G172
283     U183 = 1.0/G173
284     U184 = 1.0/G174
285     U185 = 1.0/G175
286     U186 = 1.0/G176
287     U187 = 1.0/G177
288     U188 = 1.0/G178
289     U189 = 1.0/G179
290     U190 = 1.0/G180
291     U191 = 1.0/G181
292     U192 = 1.0/G182
293     U193 = 1.0/G183
294     U194 = 1.0/G184
295     U195 = 1.0/G185
296     U196 = 1.0/G186
297     U197 = 1.0/G187
298     U198 = 1.0/G188
299     U199 = 1.0/G189
300     U200 = 1.0/G190
301     U201 = 1.0/G191
302     U202 = 1.0/G192
303     U203 = 1.0/G193
304     U204 = 1.0/G194
305     U205 = 1.0/G195
306     U206 = 1.0/G196
307     U207 = 1.0/G197
308     U208 = 1.0/G198
309     U209 = 1.0/G199
310     U210 = 1.0/G200
311     U211 = 1.0/G201
312     U212 = 1.0/G202
313     U213 = 1.0/G203
314     U214 = 1.0/G204
315     U215 = 1.0/G205
316     U216 = 1.0/G206
317     U217 = 1.0/G207
318     U218 = 1.0/G208
319     U219 = 1.0/G209
320     U220 = 1.0/G210
321     U221 = 1.0/G211
322     U222 = 1.0/G212
323     U223 = 1.0/G213
324     U224 = 1.0/G214
325     U225 = 1.0/G215
326     U226 = 1.0/G216
327     U227 = 1.0/G217
328     U228 = 1.0/G218
329     U229 = 1.0/G219
330     U230 = 1.0/G220
331     U231 = 1.0/G221
332     U232 = 1.0/G222
333     U233 = 1.0/G223
334     U234 = 1.0/G224
335     U235 = 1.0/G225
336     U236 = 1.0/G226
337     U237 = 1.0/G227
338     U238 = 1.0/G228
339     U239 = 1.0/G229
340     U240 = 1.0/G230
341     U241 = 1.0/G231
342     U242 = 1.0/G232
343     U243 = 1.0/G233
344     U244 = 1.0/G234
345     U245 = 1.0/G235
346     U246 = 1.0/G236
347     U247 = 1.0/G237
348     U248 = 1.0/G238
349     U249 = 1.0/G239
350     U250 = 1.0/G240
351     U251 = 1.0/G241
352     U252 = 1.0/G242
353     U253 = 1.0/G243
354     U254 = 1.0/G244
355     U255 = 1.0/G245
356     U256 = 1.0/G246
357     U257 = 1.0/G247
358     U258 = 1.0/G248
359     U259 = 1.0/G249
360     U260 = 1.0/G250
361     U261 = 1.0/G251
362     U262 = 1.0/G252
363     U263 = 1.0/G253
364     U264 = 1.0/G254
365     U265 = 1.0/G255
366     U266 = 1.0/G256
367     U267 = 1.0/G257
368     U268 = 1.0/G258
369     U269 = 1.0/G259
370     U270 = 1.0/G260
371     U271 = 1.0/G261
372     U272 = 1.0/G262
373     U273 = 1.0/G263
374     U274 = 1.0/G264
375     U275 = 1.0/G265
376     U276 = 1.0/G266
377     U277 = 1.0/G267
378     U278 = 1.0/G268
379     U279 = 1.0/G269
380     U280 = 1.0/G270
381     U281 = 1.0/G271
382     U282 = 1.0/G272
383     U283 = 1.0/G273
384     U284 = 1.0/G274
385     U285 = 1.0/G275
386     U286 = 1.0/G276
387     U287 = 1.0/G277
388     U288 = 1.0/G278
389     U289 = 1.0/G279
390     U290 = 1.0/G280
391     U291 = 1.0/G281
392     U292 = 1.0/G282
393     U293 = 1.0/G283
394     U294 = 1.0/G284
395     U295 = 1.0/G285
396     U296 = 1.0/G286
397     U297 = 1.0/G287
398     U298 = 1.0/G288
399     U299 = 1.0/G289
400     U300 = 1.0/G290
401     U301 = 1.0/G291
402     U302 = 1.0/G292
403     U303 = 1.0/G293
404     U304 = 1.0/G294
405     U305 = 1.0/G295
406     U306 = 1.0/G296
407     U307 = 1.0/G297
408     U308 = 1.0/G298
409     U309 = 1.0/G299
410     U310 = 1.0/G300
411     U311 = 1.0/G301
412     U312 = 1.0/G302
413     U313 = 1.0/G303
414     U314 = 1.0/G304
415     U315 = 1.0/G305
416     U316 = 1.0/G306
417     U317 = 1.0/G307
418     U318 = 1.0/G308
419     U319 = 1.0/G309
420     U320 = 1.0/G310
421     U321 = 1.0/G311
422     U322 = 1.0/G312
423     U323 = 1.0/G313
424     U324 = 1.0/G314
425     U325 = 1.0/G315
426     U326 = 1.0/G316
427     U327 = 1.0/G317
428     U328 = 1.0/G318
429     U329 = 1.0/G319
430     U330 = 1.0/G320
431     U331 = 1.0/G321
432     U332 = 1.0/G322
433     U333 = 1.0/G323
434     U334 = 1.0/G324
435     U335 = 1.0/G325
436     U336 = 1.0/G326
437     U337 = 1.0/G327
438     U338 = 1.0/G328
439     U339 = 1.0/G329
440     U340 = 1.0/G330
441     U341 = 1.0/G331
442     U342 = 1.0/G332
443     U343 = 1.0/G333
444     U344 = 1.0/G334
445     U345 = 1.0/G335
446     U346 = 1.0/G336
447     U347 = 1.0/G337
448     U348 = 1.0/G338
449     U349 = 1.0/G339
450     U350 = 1.0/G340
451     U351 = 1.0/G341
452     U352 = 1.0/G342
453     U353 = 1.0/G343
454     U354 = 1.0/G344
455     U355 = 1.0/G345
456     U356 = 1.0/G346
457     U357 = 1.0/G347
458     U358 = 1.0/G348
459     U359 = 1.0/G349
460     U360 = 1.0/G350
461     U361 = 1.0/G351
462     U362 = 1.0/G352
463     U363 = 1.0/G353
464     U364 = 1.0/G354
465     U365 = 1.0/G355
466     U366 = 1.0/G356
467     U367 = 1.0/G357
468     U368 = 1.0/G358
469     U369 = 1.0/G359
470     U370 = 1.0/G360
471     U371 = 1.0/G361
472     U372 = 1.0/G362
473     U373 = 1.0/G363
474     U374 = 1.0/G364
475     U375 = 1.0/G365
476     U376 = 1.0/G366
477     U377 = 1.0/G367
478     U378 = 1.0/G368
479     U379 = 1.0/G369
480     U380 = 1.0/G370
481     U381 = 1.0/G371
482     U382 = 1.0/G372
483     U383 = 1.0/G373
484     U384 = 1.0/G374
485     U385 = 1.0/G375
486     U386 = 1.0/G376
487     U387 = 1.0/G377
488     U388 = 1.0/G378
489     U389 = 1.0/G379
490     U390 = 1.0/G380
491     U391 = 1.0/G381
492     U392 = 1.0/G382
493     U393 = 1.0/G383
494     U394 = 1.0/G384
495     U395 = 1.0/G385
496     U396 = 1.0/G386
497     U397 = 1.0/G387
498     U398 = 1.0/G388
499     U399 = 1.0/G389
500     U400 = 1.0/G390
501     U401 = 1.0/G391
502     U402 = 1.0/G392
503     U403 = 1.0/G393
504     U404 = 1.0/G394
505     U405 = 1.0/G395
506     U406 = 1.0/G396
507     U407 = 1.0/G397
508     U408 = 1.0/G398
509     U409 = 1.0/G399
510     U410 = 1.0/G400
511     U411 = 1.0/G401
512     U412 = 1.0/G402
513     U413 = 1.0/G403
514     U414 = 1.0/G404
515     U415 = 1.0/G405
516     U416 = 1.0/G406
517     U417 = 1.0/G407
518     U418 = 1.0/G408
519     U419 = 1.0/G409
520     U420 = 1.0/G410
521     U421 = 1.0/G411
522     U422 = 1.0/G412
523     U423 = 1.0/G413
524     U424 = 1.0/G414
525     U425 = 1.0/G415
526     U426 = 1.0/G416
527     U427 = 1.0/G417
528     U428 = 1.0/G418
529     U429 = 1.0/G419
530     U430 = 1.0/G420
531     U431 = 1.0/G421
532     U432 = 1.0/G422
533     U433 = 1.0/G423
534     U434 = 1.0/G424
535     U435 = 1.0/G425
536     U436 = 1.0/G426
537     U437 = 1.0/G427
538     U438 = 1.0/G428
539     U439 = 1.0/G429
540     U440 = 1.0/G430
541     U441 = 1.0/G431
542     U442 = 1.0/G432
543     U443 = 1.0/G433
544     U444 = 1.0/G434
545     U445 = 1.0/G435
546     U446 = 1.0/G436
547     U447 = 1.0/G437
548     U448 = 1.0/G438
549     U449 = 1.0/G439
550     U450 = 1.0/G440
551     U451 = 1.0/G441
552     U452 = 1.0/G442
553     U453 = 1.0/G443
554     U454 = 1.0/G444
555     U455 = 1.0/G445
556     U456 = 1.0/G446
557     U457 = 1.0/G447
558     U458 = 1.0/G448
559     U459 = 1.0/G449
560     U460 = 1.0/G450
561     U461 = 1.0/G451
562     U462 = 1.0/G452
563     U463 = 1.0/G453
564     U464 = 1.0/G454
565     U465 = 1.0/G455
566     U466 = 1.0/G456
567     U467 = 1.0/G457
568     U468 = 1.0/G458
569     U469 = 1.0/G459
570     U470 = 1.0/G460
571     U471 = 1.0/G461
572     U472 = 1.0/G462
573     U473 = 1.0/G463
574     U474 = 1.0/G464
575     U475 = 1.0/G465
576     U476 = 1.0/G466
577     U477 = 1.0/G467
578     U478 = 1.0/G468
579     U479 = 1.0/G469
580     U480 = 1.0/G470
581     U481 = 1.0/G471
582     U482 = 1.0/G472
583     U483 = 1.0/G473
584     U484 = 1.0/G474
585     U485 = 1.0/G475
586     U486 = 1.0/G476
587     U487 = 1.0/G477
588     U488 = 1.0/G478
589     U489 = 1.0/G479
590     U490 = 1.0/G480
591     U491 = 1.0/G481
592     U492 = 1.0/G482
593     U493 = 1.0/G483
594     U494 = 1.0/G484
595     U495 = 1.0/G485
596     U496 = 1.0/G486
597     U497 = 1.0/G487
598     U498 = 1.0/G488
599     U499 = 1.0/G489
600     U500 = 1.0/G490
601     U501 = 1.0/G491
602     U502 = 1.0/G492
603     U503 = 1.0/G493
604     U504 = 1.0/G494
605     U505 = 1.0/G495
606     U506 = 1.0/G496
607     U507 = 1.0/G497
608     U508 = 1.0/G498
609     U509 = 1.0/G499
610     U510 = 1.0/G500
611     U511 = 1.0/G501
612     U512 = 1.0/G502
613     U513 = 1.0/G503
614     U514 = 1.0/G504
615     U515 = 1.0/G505
616     U516 = 1.0/G506
617     U517 = 1.0/G507
618     U518 = 1.0/G508
619     U519 = 1.0/G509
620     U520 = 1.0/G510
621     U521 = 1.0/G511
622     U522 = 1.0/G512
623     U523 = 1.0/G513
624     U524 = 1.0/G514
625     U525 = 1.0/G515
626     U526 = 1.0/G516
627     U527 = 1.0/G517
628     U528 = 1.0/G518
629     U529 = 1.0/G519
630     U530 = 1.0/G520
631     U531 = 1.0/G521
632     U532 = 1.0/G522
633     U533 = 1.0/G523
634     U534 = 1.0/G524
635     U535 = 1.0/G525
636     U536 = 1.0/G526
637     U537 = 1.0/G527
638     U538 = 1.0/G528
639     U539 = 1.0/G529
640     U540 = 1.0/G530
641     U541 = 1.0/G531
642     U542 = 1.0/G532
643     U543 = 1.0/G533
644     U544 = 1.0/G534
645     U545 = 1.0/G535
646     U546 = 1.0/G536
647     U547 = 1.0/G537
648     U548 = 1.0/G538
649     U549 = 1.0/G539
650     U550 = 1.0/G540
651     U551 = 1.0/G541
652     U552 = 1.0/G542
653     U553 = 1.0/G543
654     U554 = 1.0/G544
655     U555 = 1.0/G545
656     U556 = 1.0/G546
657     U557 = 1.0/G547
658     U558 = 1.0/G548
659     U559 = 1.0/G549
660     U560 = 1.0/G550
661     U561 = 1.0/G551
662     U562 = 1.0/G552
663     U563 = 1.0/G553
664     U564 = 1.0/G554
665     U565 = 1.0/G555
666     U566 = 1.0/G556
667     U567 = 1.0/G557
668     U568 = 1.0/G558
669     U569 = 1.0/G559
670     U570 = 1.0/G560
671     U571 = 1.0/G561
672     U572 = 1.0/G562
673     U573 = 1.0/G563
674     U574 = 1.0/G564
675     U575 = 1.0/G565
676     U576 = 1.0/G566
677     U577 = 1.0/G567
678     U578 = 1.0/G568
679     U579 = 1.0/G569
680     U580 = 1.0/G570
681     U581 = 1.0/G571
682     U582 = 1.0/G572
683     U583 = 1.0/G573
684     U584 = 1.0/G574
685     U585 = 1.0/G575
686     U586 = 1.0/G576
687     U587 = 1.0/G577
688     U588 = 1.0/G578
689     U589 = 1.0/G579
690     U590 = 1.0/G580
691     U591 = 1.0/G581
692     U592 = 1.0/G582
693     U593 = 1.0/G583
694     U594 = 1.0/G584
695     U595 = 1.0/G585
696     U596 = 1.0/G586
697     U597 = 1.0/G587
698     U598 = 1.0/G588
699     U599 = 1.0/G589
700     U600 = 1.0/G590
701     U601 = 1.0/G591
702     U602 = 1.0/G592
703     U603 = 1.0/G593
704     U604 = 1.0/G594
705     U605 = 1.0/G595
706     U606 = 1.0/G596
707     U607 = 1.0/G597
708     U608 = 1.0/G598
709     U609 = 1.0/G599
710     U610 = 1.0/G600
711     U611 = 1.0/G601
712     U612 = 1.0/G602
713     U613 = 1.0/G603
714     U614 = 1.0/G604
715     U615 = 1.0/G605
716     U616 = 1.0/G606
717     U617 = 1.0/G607
718     U618 = 1.0/G608
719     U619 = 1.0/G609
720     U620 = 1.0/G610
721     U621 = 1.0/G611
722     U622 = 1.0/G612
723     U623 = 1.0/G613
724     U624 = 1.0/G614
725     U625 = 1.0/G615
726     U626 = 1.0/G616
727     U627 = 1.0/G617
728     U628 = 1.0/G618
729     U629 = 1.0/G619
730     U630 = 1.0/G620
731     U631 = 1.0/G621
732     U632 = 1.0/G622
733     U633 = 1.0/G623
734     U634 = 1.0/G624
735     U635 = 1.0/G625
736     U636 = 1.0/G626
737     U637 = 1.0/G627
738     U638 = 1.0/G628
739     U639 = 1.0/G629
740     U640 = 1.0/G630
741     U641 = 1.0/G631
742     U642 = 1.0/G632
743     U643 = 1.0/G633
744     U644 = 1.0/G634
745     U645 = 1.0/G635
746     U646 = 1.0/G636
747     U647 = 1.0/G637
748     U648 = 1.0/G638
749     U649 = 1.0/G639
750     U650 = 1.0/G640
751     U651 = 1.0/G641
752     U652 = 1.0/G642
753     U653 = 1.0/G643
754     U654 = 1.0/G644
755     U655 = 1.0/G645
756     U656 = 1.0/G646
757     U657 = 1.0/G647
758     U658 = 1.0/G648
759     U659 = 1.0/G649
760     U660 = 1.0/G650
761     U661 = 1.0/G651
762     U662 = 1.0/G652
763     U663 = 1.0/G653
764     U664 = 1.0/G654
765     U665 = 1.0/G655
766     U666 = 1.0/G656
767     U667 = 1.0/G657
768     U668 = 1.0/G658
769     U669 = 1.0/G659
770     U670 = 1.0/G660
771     U671 = 1.0/G661
772     U672 = 1.0/G662
773     U673 = 1.0/G663
774     U674 = 1.0/G664
775     U675 = 1.0/G665
776     U676 = 1.0/G666
777     U677 = 1.0/G667
778     U678 = 1.0/G668
779     U679 = 1.0/G669
7
```

```

036      C4*U4*G4C1)*      END$
037      END$
038      NU = NU1 $
039      COMMENT CHECK THE SIDE CONSTRAINTS $
040      G3 = DMAX - CON1 $      G35 = G3*G3 $
041      G4 = TMAX - CON2 $      G45 = G4*G4 $
042      IF G3C LEQ 0.0 OR G4C LEQ 0.0 THEN BEGIN
043      ALT = 4 $
044      GO TO RETREAT $ END$
045      TS = TS + NU/G35 $
046      PHEE = PHEE + NU*(1.0/G3 + 1.0/G4) $
047      TS = TS + NU/G35 $
048      TS1=TS1+NU/G45 $
049      IF B4 THEN BEGIN
050      AAA = AAA + 2.0*NU/G3C $
051      HBR = HBR + 2.0*NU/G4C $
052      END$
053      CON3 = CON1/CON2 $
054      G1 = CON3 - RAL $
055      G2 = RAD - CON3 $
056      G1 = SUS(G1*-13) $      G2 = SUS(G2*-13) $
057      IF G1 LEQ 0.0 OR G2 LEQ 0.0 THEN BEGIN
058      ALT = 1 $
059      GO TO RETREAT $
060      CON3 = CON3/CON2 $
061      CON4 = CON3 * CON3 $
062      CON5 = CON3/CON2 $
063      G15 = G1*G1 $
064      G25 = G2*G2 $
065      G1L = G1*G15 $
066      G2C = G2*G25 $
067      PHEE = PHEE + NU*( 1.0/G1 + 1.0/G2 ) $
068      TS = TS + NU*( 1/G25 - 1/G15)/CON2 $
069      TS1=TS1+ NU*(1/G15 - 1/G25)*CON3 $
070      IF B4 THEN BEGIN
071      AAA = AAA + NU*(2/G2C - 2/G15)*CON5 + (2/G1L + 2/G2C)*CON4 $
072      HBR = HBR + NU*( 2/G25 - 2/G15)*CON5 + (2/G1L + 2/G2C)*CON4 $
073      END$
074      CON1 = SUS(CON1*-13) $
075      IF CON1 LEQ 0.0 THEN BEGIN ALT = 3 $      GO TO RETREAT $ END$
076      PHEE = PHEE + NU/CON1 $
077      IF B3 THEN TS = TS - NU/( CON1*CON1) $
078      IF B4 THEN AAA = AAA + 2*NU/( CON1*CON1*CON1) $
079      CON6 = NEW $
080      CON7 = CON6*W $
081      G1 = MESS * W/(1)/CON6 $
082      ALPH = CON1 $
083      BET = CON2 $
084      TS = TS + G1*HET $
085      TS1= TS1 + G1*ALPH $
086      IF B4 THEN BEGIN
087      CON9 = 2*MESS*W/(1)*W/(1)/CON7 $
088      IF 10 NEG 0 THEN
089      TS=SCALE( NKT+ID ) = TS*SCALE( NKT + ID ) + AAA *CON9*HET*BET $

```

```

b90      IF IT NEQ 0 THEN
b91          TSCALE( NKTD+1 ) = TSCALE( NKTD+1 ) + BDB + CONY*ALPH*ALPH $
b92      ENDS
b93      IF ID NEQ 0 THEN DELX( NKTD + 1 ) = DELX( NKTD + 1 ) + TS $
b94      IF IT NEQ 0 THEN DELX( NKTD + 1 ) = DELX( NKTD + 1 ) + TS1 $
b95      ENDS
b96      SAVPH1 = PHEE $
b97      LIR..
b98      IF B10 THEN BEGIN
b99          CHEK = 4 $
700          W = WU - W $
701          B10 = FALSE $
702          GO TO L19 $ ENDS
703      IF B4 THEN BEGIN
704          FOR K = (1,1,NKTD) DO A(K) = X(K)*TSCALE(K) $
705          IF NOT B5 THEN FOR I = (NKTD+1,NKTD) DO TSCALE(I) = 1.0 $
706          FOR I = (1,1,NKTD) DO TSCALE(I) = 1.0/SQR( ABS(TSCALE(I)) ) $
707          G1 = MAX( TSCALE ) $
708          FOR I = (1,1,NKTD) DO ISCALF(I) = TSCALE(I)/G1 $
709          IF NOT B5 THEN FOR I = (1 + NKTD,NKTD) DO TSCALE(I) = 1.0 $
710          IF CHEK EQL 4 OR B6 THEN GO TO L3R $
711          FOR I = (1,1,NKTD) DO
712              IF ABS( TSCALE(I) - TSCALE(I)*TSCALE(I) ) GTR 0.60 THEN GO TO
713                  LSK $
714              DELX(I) = 8.0 $
715              FOR I = (1,1,NKTD) DO TSCALE(I) = TSCALE(I) $
716              LSR..
717          WRITE (1,1,NKTD) DO A(I) = X(I)/TSCALE(I) $
718          ENDS
719      IF CHEK EQL 43 OR CHEK EQL 4 THEN WRITE (F1A, NU,THG,MM,MO,X,TSCALE) $
720      CTJ = CTJ + 1 $
721      IF CTJ GEQ 35 THEN BEGIN
722          CTJ = 0 $
723          WRITE (1,1,NKTD) $
724          WRITE (F1A,N*MO-W) $
725          ENDS
726      RETREAT..
727      IF B6 THEN BEGIN
728          WRITE (EJ) $
729          WRITE (1,1,NKTD) $
730          V = W/5 $
731          FOR I = (1,1,NKTD) DO BEGIN
732              IF I LEQ V THEN L = 5 ELSE L = M - 5*V $
733              B = 5*(1-I) $
734              WRITE (F1A,FUR J = (1,1,L) DO B + J) $
735              WRITE (F1A,FUR J = (1,1,L) DO SMAX(B+J) ) $
736              WRITE (F1A,FUR J = (1,1,L) DO SMIN(B+J) ) $
737              WRITE (F1A,FUR J = (1,1,L) DO -STRESS(B+J,N) ) $
738              WRITE (F1A,FUR J = (1,1,L) DO -STRESS(B+J,-1) ) $
739              WRITE (1,1,NKTD) $
740              FOR K = (1,1,K1) DO
741                  WRITE (F1A,K,FUR J = (1,1,L) DO STRESS( B+J,K) ) $
742                  ENDS
743      ENDS

```

```

284      WRITE (EJ)$
285      IF B6 THEN ALT = 8 $
286      IF B8 THEN BEGIN
287      WRITE ('THE LAST BALANCED DESIGN WILL BE ACCEPTED AS THE OPTIMUM')$
288      DELX(0) = 101.0 $
289      END$
290      IF B6 AND B7 THEN BEGIN
291      WRITE (PUNCH,NU,THG,WM,MO,X,TSCALE)$
292      WRITE (
293      IF ALT EQL 0 OR ALT EWL 0 THEN
294      FOR I = (1,1,NAUT) DO BEGIN
295      DELX(I) = DELX(1)*TSCALE(1)$
296      END$
297      ROM = ALT $
298      PHEE = PHEE + N + MESS / W $
299      END$
300      END$
301      END$
302      END$
303      END$
304      END$
305      END$
306      END$
307      END$
308      END$
309      END$
310      END$
311      END$
312      END$
313      END$
314      END$
315      END$
316      END$
317      END$
318      END$
319      END$
320      END$
321      END$
322      END$
323      END$
324      END$
325      END$
326      END$
327      END$
328      END$
329      END$
330      END$
331      END$
332      END$
333      END$
334      END$
335      END$
336      END$
337      END$
338      END$
339      END$
340      END$
341      END$
342      END$
343      END$
344      END$
345      END$
346      END$
347      END$
348      END$
349      END$
350      END$
351      END$
352      END$
353      END$
354      END$
355      END$
356      END$
357      END$
358      END$
359      END$
360      END$
361      END$
362      END$
363      END$
364      END$
365      END$
366      END$
367      END$
368      END$
369      END$
370      END$
371      END$
372      END$
373      END$
374      END$
375      END$
376      END$
377      END$
378      END$
379      END$
380      END$
381      END$
382      END$
383      END$
384      END$
385      END$
386      END$
387      END$
388      END$
389      END$
390      END$
391      END$
392      END$
393      END$
394      END$
395      END$
396      END$
397      END$
398      END$
399      END$
400      END$
401      END$
402      END$
403      END$
404      END$
405      END$
406      END$
407      END$
408      END$
409      END$
410      END$
411      END$
412      END$
413      END$
414      END$
415      END$
416      END$
417      END$
418      END$
419      END$
420      END$
421      END$
422      END$
423      END$
424      END$
425      END$
426      END$
427      END$
428      END$
429      END$
430      END$
431      END$
432      END$
433      END$
434      END$
435      END$
436      END$
437      END$
438      END$
439      END$
440      END$
441      END$
442      END$
443      END$
444      END$
445      END$
446      END$
447      END$
448      END$
449      END$
450      END$
451      END$
452      END$
453      END$
454      END$
455      END$
456      END$
457      END$
458      END$
459      END$
460      END$
461      END$
462      END$
463      END$
464      END$
465      END$
466      END$
467      END$
468      END$
469      END$
470      END$
471      END$
472      END$
473      END$
474      END$
475      END$
476      END$
477      END$
478      END$
479      END$
480      END$
481      END$
482      END$
483      END$
484      END$
485      END$
486      END$
487      END$
488      END$
489      END$
490      END$
491      END$
492      END$
493      END$
494      END$
495      END$
496      END$
497      END$
498      END$
499      END$
500      END$
501      END$
502      END$
503      END$
504      END$
505      END$
506      END$
507      END$
508      END$
509      END$
510      END$
511      END$
512      END$
513      END$
514      END$
515      END$
516      END$
517      END$
518      END$
519      END$
520      END$
521      END$
522      END$
523      END$
524      END$
525      END$
526      END$
527      END$
528      END$
529      END$
530      END$
531      END$
532      END$
533      END$
534      END$
535      END$
536      END$
537      END$
538      END$
539      END$
540      END$
541      END$
542      END$
543      END$
544      END$
545      END$
546      END$
547      END$
548      END$
549      END$
550      END$
551      END$
552      END$
553      END$
554      END$
555      END$
556      END$
557      END$
558      END$
559      END$
560      END$
561      END$
562      END$
563      END$
564      END$
565      END$
566      END$
567      END$
568      END$
569      END$
570      END$
571      END$
572      END$
573      END$
574      END$
575      END$
576      END$
577      END$
578      END$
579      END$
580      END$
581      END$
582      END$
583      END$
584      END$
585      END$
586      END$
587      END$
588      END$
589      END$
590      END$
591      END$
592      END$
593      END$
594      END$
595      END$
596      END$
597      END$
598      END$
599      END$
600      END$
601      END$
602      END$
603      END$
604      END$
605      END$
606      END$
607      END$
608      END$
609      END$
610      END$
611      END$
612      END$
613      END$
614      END$
615      END$
616      END$
617      END$
618      END$
619      END$
620      END$
621      END$
622      END$
623      END$
624      END$
625      END$
626      END$
627      END$
628      END$
629      END$
630      END$
631      END$
632      END$
633      END$
634      END$
635      END$
636      END$
637      END$
638      END$
639      END$
640      END$
641      END$
642      END$
643      END$
644      END$
645      END$
646      END$
647      END$
648      END$
649      END$
650      END$
651      END$
652      END$
653      END$
654      END$
655      END$
656      END$
657      END$
658      END$
659      END$
660      END$
661      END$
662      END$
663      END$
664      END$
665      END$
666      END$
667      END$
668      END$
669      END$
670      END$
671      END$
672      END$
673      END$
674      END$
675      END$
676      END$
677      END$
678      END$
679      END$
680      END$
681      END$
682      END$
683      END$
684      END$
685      END$
686      END$
687      END$
688      END$
689      END$
690      END$
691      END$
692      END$
693      END$
694      END$
695      END$
696      END$
697      END$
698      END$
699      END$
700      END$
701      END$
702      END$
703      END$
704      END$
705      END$
706      END$
707      END$
708      END$
709      END$
710      END$
711      END$
712      END$
713      END$
714      END$
715      END$
716      END$
717      END$
718      END$
719      END$
720      END$
721      END$
722      END$
723      END$
724      END$
725      END$
726      END$
727      END$
728      END$
729      END$
730      END$
731      END$
732      END$
733      END$
734      END$
735      END$
736      END$
737      END$
738      END$
739      END$
740      END$
741      END$
742      END$
743      END$
744      END$
745      END$
746      END$
747      END$
748      END$
749      END$
750      END$
751      END$
752      END$
753      END$
754      END$
755      END$
756      END$
757      END$
758      END$
759      END$
760      END$
761      END$
762      END$
763      END$
764      END$
765      END$
766      END$
767      END$
768      END$
769      END$
770      END$
771      END$
772      END$
773      END$
774      END$
775      END$
776      END$
777      END$
778      END$
779      END$
780      END$
781      END$
782      END$
783      END$
784      END$
785      END$
786      END$
787      END$
788      END$
789      END$
790      END$
791      END$
792      END$
793      END$
794      END$
795      END$
796      END$
797      END$
798      END$
799      END$
800      END$
801      END$
802      END$
803      END$
804      END$
805      END$
806      END$
807      END$
808      END$
809      END$
810      END$
811      END$
812      END$
813      END$
814      END$
815      END$
816      END$
817      END$
818      END$
819      END$
820      END$
821      END$
822      END$
823      END$
824      END$
825      END$
826      END$
827      END$
828      END$
829      END$
830      END$
831      END$
832      END$
833      END$
834      END$
835      END$
836      END$
837      END$
838      END$
839      END$
840      END$
841      END$
842      END$
843      END$
844      END$
845      END$
846      END$
847      END$
848      END$
849      END$
850      END$
851      END$
852      END$
853      END$
854      END$
855      END$
856      END$
857      END$
858      END$
859      END$
860      END$
861      END$
862      END$
863      END$
864      END$
865      END$
866      END$
867      END$
868      END$
869      END$
870      END$
871      END$
872      END$
873      END$
874      END$
875      END$
876      END$
877      END$
878      END$
879      END$
880      END$
881      END$
882      END$
883      END$
884      END$
885      END$
886      END$
887      END$
888      END$
889      END$
890      END$
891      END$
892      END$
893      END$
894      END$
895      END$
896      END$
897      END$
898      END$
899      END$
900      END$
901      END$
902      END$
903      END$
904      END$
905      END$
906      END$
907      END$
908      END$
909      END$
910      END$
911      END$
912      END$
913      END$
914      END$
915      END$
916      END$
917      END$
918      END$
919      END$
920      END$
921      END$
922      END$
923      END$
924      END$
925      END$
926      END$
927      END$
928      END$
929      END$
930      END$
931      END$
932      END$
933      END$
934      END$
935      END$
936      END$
937      END$
938      END$
939      END$
940      END$
941      END$
942      END$
943      END$
944      END$
945      END$
946      END$
947      END$
948      END$
949      END$
950      END$
951      END$
952      END$
953      END$
954      END$
955      END$
956      END$
957      END$
958      END$
959      END$
960      END$
961      END$
962      END$
963      END$
964      END$
965      END$
966      END$
967      END$
968      END$
969      END$
970      END$
971      END$
972      END$
973      END$
974      END$
975      END$
976      END$
977      END$
978      END$
979      END$
980      END$
981      END$
982      END$
983      END$
984      END$
985      END$
986      END$
987      END$
988      END$
989      END$
990      END$
991      END$
992      END$
993      END$
994      END$
995      END$
996      END$
997      END$
998      END$
999      END$
1000     END$

```

COMPILATION COMPLETED IN 34.76 SECONDS



```

ALU      FLME
COMPILED BY CASE 1107 ALGOL 60 (FAST VERSION) DATED DECEMBER 14, 1967
THIS COMPILATION WAS DONE ON 09 APR 68 AT 14151842
80      EXTERNAL PROCEDURE FP S
BLOCK 1  LEVEL 1
2      INTEGER NO,NT,N,M,NM,NOV,NTVS
3      READ (KT,N,M,NM,NOV,NTVS)
4      NC = N S
5      N = NOEKT + NOV + NYV S
6      BEGIN
7      COMMENT FLEICHER - DEEVE'S S
8      PROCEDURE SGENI(N,S,GX,UYPH)S
9
10     REAL ARRAY S,GX,GXEM S
11     INTEGER N S
12     BEGIN
13     REAL BETA,BETAS
14     INTEGER IS
15     BETA = 0.0 S
16     FOR I = (1,1,N) DO BEGIN
17     BE = PV + GXEM(I)*GXEM(I)S
18     TA = TA + GX(I)*GX(I)S
19     ENDS
20     BETA = BE/TA S
21     FOR I = (1,1,N) DO S(I) = GXEM(I) + BETA*S(I)S
22     ENDS
23
24     REAL ARRAY A1U,GXEM,XEM,GAB,S,TGA,XA(10)S
25     REAL O1,O2,O3,M1,M2,M3, PA, EPA,EPR,EG, FI, T, FEM, GG,
26     TS, A, B, FAI, FB, WAT, GS, GS, AGEM, AG, TEST,
27     M, GXS, FST, TIME, CTIME, CG, TO, TM, TZ, TC,
28     T2, G2, TJN, TSA, TEA, TA, Z, N, TGAS S
29     LOCAL LABEL L5A, LPTAP S
30     INTEGER O1, RELOCS, RO, CHECK, CTAL, CTMB, CTNC, CTND, CTNE S
31     BOOLEAN B1,B2,B3,B4,B5,B6,B7
32     FORMAT FORAL M16,B,11S
33     TIME = CTIME = CLOK S
34     CHECK = 43S
35     B1 = B2 = B3 = TRUE S
36     B4 = FALSE S
37     O1 = O2 = O3 = M1 = M2 = M3 = 0.0 S
38     EPB = 1.0E-4 S
39     EG = 0.0 S
40     G6 = 1.0 S
41     FP( RO,PA,FB,FI,B1,B2,B3,CHECK )S
42     READ (CTRC)S
43     READ ( B5 )S
44     CTMB = CTNC S
45     O = NS
46     B1 = FALSE S
47     WRITE ( CHECK, 'PA', X, ' ', FI, ' ', GXS

```

```

47      UPTOP1
48      TS = 0.0 S
49      FOR I = (1,1:N) DO BEGIN
50          S(I) = G(I) S
51          TS = TS + S(I) S(I) S
52      ENDS
53      S(0) = SORT( TS ) S
54      ENDS
55
56      TOP1
57      FEST = 0.0 S
58      CTRA = 0 S
59      CHECK = 1 S
60      Q = 0 S
61
62      L21
63      IF NOT B5 THEN
64          IF CTRA GEC CTRB THEN BEGIN
65              CHECK = 2 S
66              FM ( 208,X1,G1,1,81,82,83,CHECK ) S
67              IF G(1) EQL 8.0 THEN BEGIN
68                  G(0) = 0.0 S
69                  CTRB = 2*CTRA S
70                  GO TO TOP S
71              ENDS
72              CTRU = CTRD + 1 S
73              IF MOD( CTRD,3 ) EQL 0 THEN BEGIN
74                  CTRD = 0 S
75                  CTRB = CTRA + 1 S
76                  GO TO TOP S
77              ENDS
78              GO TO UPTOP S
79          ENDS
80
81      TS = 0.0 S
82      FOR I = (1,1:N) DO
83          TS = TS + G(I) S(I) S
84      ENDS
85      IF GG LSS 0.0 THEN BEGIN
86          FOR I = (1,1:N) DO S(I) = G(I) S
87          WRITE (15) IS REPLACED BY 1, BECAUSE G DOT S IS NEGATIVE S
88          GO TO UPTOP S
89      ENDS
90
91      L41
92      RELOCS = 0 S
93      TS = F1 - FEST S
94
95      100

```

```

101 IF TS LSS 0.0 THEN BEGIN
102   FEST = FEST + 2.0E15 $
103   GO TO L48
104   ENDS
105 GAS = -66 $
106 IF GC EGL 0.0 THEN GO = 1.0 $
107 T = 2.0EQUATS/GC $
108 TC = T $
109 R4 = FALSE $
110 CTRA = CTRA + 1 $
111 TWIN = 0.0 $
112 T2 = 0.0 $
113 G2 = GAS $
114 WRITE (G2) $
115 B7 = FALSE $
116
117
118 UPPER1
119 M = T $
120 B6 = TRUE $
121
122
123
124
125 INCFASE1
126 IF B7 AND M GEQ TWIN THEN BEGIN
127   NOS = 1 $
128   GO TO *105
129   ENDS
130 FOR I = (1.1N) TO XX(I) = X(I) - NOS(I) $
131   E = M $
132   F(I) = X(I) - G(I) * FB(I) * P2 * B * C * H * K $
133
134
135 MID1
136 WRITE (M * NOS * FB) $
137 IF NOS NEQ 1 THEN BEGIN
138   IF NOS EQL 5 THEN BEGIN
139     WRITE ('RESIDUALS WERE SATISFIED IN PART 1 OF LIN SEARCH') $
140     F1 = FB $
141     B2 = FALSE $
142     CTR0 = 0 $
143     CYRB = CTRC $
144     FOR I = (1.1N) DO BEGIN
145       G(I) = G(I) $
146       X(I) = X(I) $
147       ENDS
148     GO TO *105
149     ENDS
150   B7 = TRUE $
151   IF TWIN EGL 0.0 THEN TWIN = M $
152   IF M LSS TWIN THEN TWIN = M $
153   IF P6 THEN BEGIN
154     T = T / 2.0 $
155     GO TO UPPER1 AND ELSE BEGIN

```

```

155 M = TA + (M - TA)/2.0 $
156 GO TO INCREASES $
157 ENDS
158
159 TSA = TS = 0.0 $
160 FOR I = (1,1,N) DO BEGIN
161   TS = TS - GBS(I) $
162   TSA = TSA + GBS(I)*G(I) $
163   ENDS
164 GBS = TS $
165 TSA = SORT( TSA ) $
166 WRITE (GPS,TSA) $
167 IF ABS(GBS/(1.0)*TSA) > 1.0 THEN BEGIN
168   WRITE ('LINEAR MINIMUM IS FOUND IN PART 1 OF LINEAR SEARCH') $
169   WRITE (GBS/(1.0)*TSA) $
170   T = M $
171   GO TO L54 $
172   ENDS
173 IF R7 AND ABS( (1 - M)/M ) > 1.0 THEN BEGIN
174   WRITE ('TIGHT FIT INDEED: TA = M') $
175   GO TO L54 $
176   ENDS
177 IF GBS > 0.0 THEN BEGIN
178   A = 0.0 $
179   FOR I = (1,1,N) DO GA(I) = G(I) $
180   FA = F1 $
181   GO TO RELOC $
182   ENDS
183 BEGIN
184   A = TA $
185   FA = TFA $
186   FOR I = (1,1,N) DO GA(I) = TGA(I) $
187   GAS = TGAS $
188   GO TO RELOC $
189   ENDS
190 T = M $
191 TA = M $
192 TFA = FA $
193 M = MAX(1.0,1.0*GBS*(1.0 - M)/(GBS - G2) + M + T) $
194 R0 = FALSE $
195 T2 = TA $
196 G2 = GBS $
197 FOR I = (1,1,N) DO TGA(I) = GBS(I) $
198 TGAS = GPS $
199 GO TO INCREASES
200
201 RELOC
202 IF R0 THEN BEGIN
203   WRITE ('USING SLOPE INTERCEPT REFIT') $
204   R0 = FALSE $
205   T = ABS(GAS*(R - A)/(ABS(GAS) + ABS(GBS)) + A) $
206   GO TO L48 $
207   ENDS
208 Z = 3.0*(FA - FB)/(B - A) + GAS + GMS $
209 W = SORT( Z - Z = GAS*GBS ) $

```

```

209 Y = B - ((GBS + B - Z)/(GBS - GAS + 2.0*B))*(B-A))$
210
211
212
213 L481
214 FOR I = (1,1,N) DO XEM(I) = X(I) - T*G(I)$
215 FPI(ROB,XEM,GXEM,FEM,GBI,GB2,GB3,CHEK) $
216 IF NOW EOL THEN BEGIN
217   FOR I = (1,1,N) DO BEGIN
218     X(I) = XEM(I)$
219     G(I) = GXEM(I)$ ENDS
220     WRITE('RESIDUALS ARE SATISFIED IN PART 2 OF LINEAR SEARCH')$
221     F1 = FEM $
222     CTRO = 0 $
223     CTRO = CTRO $
224     BS = FALSE $
225     GO TO UPTOP $
226     ENDS
227
228
229
230 GS = AGXEM = AG = 0.0 $
231 FOR I = (1,1,N) NO BEGIN
232   GS = GS - GXEM(I)$
233   AGXEM = AGXEM + GXEM(I)*GXEM(I)$
234   ENDS
235 TEST = GS / (S10)*SOR((AGXEM) ) $
236 WRITE('FORA-T-FFM-TEST-BS')$
237 IF FPM GTR MAX ( FA,FB) CP ABS(TEST) GTR 1.0E-2 THEN BEGIN
238   RELOCS = RELOCS + 1 $
239   IF RELOCS FOL = THEN GO TO L5 $
240   TS = MIN(FA,FB,F1)$
241   IF ABS( (FPM - TS)/FPM) LSS 5.0E-7 THEN B4 = TRUE $
242   IF GS LSS 0.0 THEN BEGIN
243     FA = FPM $
244     A = T $
245     GAS = GS $
246     FOR I = (1,1,N) DO GA(I) = GXEM(I)$
247     GO TO RELOC $ FND ELSE
248     BEGIN
249       FB = FEM $
250       B = T $
251       GBS = GS $
252       FOR I = (1,1,N) DO GB(I) = GXEM(I)$
253       GO TO RELOC $
254     ENDS
255     GO TO L4 $
256
257
258
259 L5A1
260 FOR I = (1,1,N) NO BEGIN
261   XEM(I) = XX(I)$
262   GXEM(I) = GB(I)$

```

```

263      ENCS
264      AGNEW = TSA S
265      FC = FB S
266
267
268
269
270      L31
271      WRITE (XEM)13 WRITE (FUP) XEM13
272      WRITE (XEM)13 WRITE (FOR) FEM13
273      WRITE (XEM)13 WRITE (FOR) FEM13
274      QC = CTRACCTRACTZ/1 TUE(CTRA + 1100213
275      TA = TC = 0.0 S
276      FOR I = (1,14) DO
277      IF ABS( XEM(I) ) GIP TM THEN
278      T4 = IF1 - FEM13*0.087 S
279      T5 = SQRT ( AGNEW ) S
280      FOR I = (1,14) DO
281      IF ABS( XEM(I) ) - X(I) 10G(11)TM 0TM 1D THEN
282      T6 = T5*( XEM(I) - X(I) ) / T4 S
283      WRITE (FOWA) T4 + TC + T7 + F1 S
284      IF TS LSS 1.00E-6 OR
285      TC LSS 3.00E-8 CP
286      T2 LSC F1 THEN BEGIN
287      GO TO L3 S
288      ENCS
289      IF B5 THEN GO TO L6 S
290
291
292
293      CURRENT AVERAGE OF THE LAST THREE VALUES OF TU COMPARED TO FRACTION
294      OF MAXIMUM TO ENCOUNTERED S
295      IF D2 EQL 0.0 THEN M1 = P2 + TC S
296      D3 = D2 S
297      D4 = D1 S
298      D5 = TC S
299      FFA = MAX1 1001.00E-01EPA13
300      IF (D1 + D2 + D3)/%0.0 LFC EPA THEN GO TO L3 S
301
302
303
304
305      CURRENT AVERAGE OF LAST THREE PERCENT CHANGE OF THE FUNCTION VALUE F
306      COMPARED TO MAXIMUM PERCENT CHANGE, LIMITED BY 0.0001 AND 0.000023
307      EA = (F - FEM1)/FEM S
308
309      IF EX LSS 0.000075 THEN LT0E = CTR0L + 1 ELSE CTR0E = " S
310      IF CTR0E REQ 0 THEN GO TO L35
311
312      IF M2 EQL 0.0 THEN M1 = M2 = EX S
313      M3 = M2 S
314      M4 = M1 S
315      M1 = EX S
316      EG = MAX1 EUGER0100E-310

```

```

317 EPB = MIN(EPB, F1)S
318 EPB = MAX(EPB, 2.04 - S)S
319 IF M1 + M2 + M3 LESS 3.0*EPB THEN GO TO L3 S
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

EPB = MIN(EPB, F1)S  
 EPB = MAX(EPB, 2.04 - S)S  
 IF M1 + M2 + M3 LESS 3.0\*EPB THEN GO TO L3 S  
  
 L3  
 WRITE (TIME FOR CYCLE, CLOC - CTIME)  
 WRITE (TIME ELAPSED, CLOC - TIME)S  
 CTIME = CLOC S  
 SUM1 (N1 S, G + G\*EM1)  
 TS = 0.0 S  
 FOR I = (1, N1) DO BEGIN  
 TS = TS + S(I)\*S(I)S  
 G(I) = G\*EM(I)S  
 X(I) = X\*EM(I)S  
 ENDS  
 S(I) = S\*EM(I)S  
 F1 = F\*EM1  
 GO TO L2S  
  
 L31  
 WRITE (MINIMUM, X\*F\*EM1)S  
 WRITE (TIME, CLOC - TIME)S  
 M5 = FALSE S  
 CHECK = 0 S  
 WRITE (PCMA, PA, PM, PE)S  
 FOR I = (1, N1) DO X(I) = X\*EM(I)S  
 N1 = 02 = 03 = M1 = M2 = M3 = 0.0 S  
 EPB = 1.06 - 05  
 EG = EPA = 0.0 S  
 CL = 1.0 S  
 CIRC = 0 S  
 CIRC = CIRC S  
 FPI POS, 2, G, F1, M1, E2, M3, CHECK S  
 IF G(0) EOL 101.0 THEN GO TO COMPLETE  
 IF AND EOL 1 THEN BEGIN  
 WRITE (FOR, F1)S  
 MOV = 0.0 S  
 GO TO UPTOP S  
 ENDS  
 CHECK = 0.0 S  
 FPI POS, 2, G, F1, M1, E2, M3, CHECK S  
 WRITE (PCMA, F1)S  
 GO TO UPTOP S  
  
 COMPLETE  
 ENDS  
  
 L1  
 END BLOCK 2 LEVEL 2  
 248